Digital Pen Features Predict Task Difficulty and User Performance of Cognitive Tests

Michael Barz*
michael.barz@dfki.de
German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany

Kristin Altmeyer
kristin.altmeyer@uni-saarland.de
Saarland University
Saarbrücken, Germany

Sarah Malone
s.malone@mx.uni-saarland.de
Saarland University
Saarbrücken, Germany

Luisa Lauer
luisa.lauer@uni-saarland.de
Saarland University
Saarbrücken, Germany

Daniel Sonntag
daniel.sonntag@dfki.de
German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany

ABSTRACT

Digital pen signals were shown to be predictive for cognitive states, cognitive load and emotion in educational settings. We investigate whether low-level pen-based features can predict the difficulty of tasks in a cognitive test and the learner's performance in these tasks, which is inherently related to cognitive load, without a semantic content analysis. We record data for tasks of varying difficulty in a controlled study with children from elementary school. We include two versions of the Trail Making Test (TMT) and six drawing patterns from the Snijders-Oomen Non-verbal intelligence test (SON) as tasks that feature increasing levels of difficulty. We examine how accurately we can predict the task difficulty and the user performance as a measure for cognitive load using support vector machines and gradient boosted decision trees with different feature selection strategies. The results show that our correlation-based feature selection is beneficial for model training, in particular when samples from TMT and SON are concatenated for joint modelling of difficulty and time. Our findings open up opportunities for technology-enhanced adaptive learning.

CCS CONCEPTS

• **Human-centered computing** → **User models**; *Human computer interaction (HCI)*; *Interaction design*; • **Computing methodologies** → *Machine learning*; *Active learning settings*; • **Applied computing** → Education.

KEYWORDS

digital pen; pen interaction; cognitive tests; education; machine learning; feature selection

*PhD student at the Saarbrücken Graduate School of Computer Science

UMAP '20, July 14–17, 2020, Genoa, Italy

ACM ISBN 978-1-4503-6861-2/20/07…$15.00
https://doi.org/10.1145/3340631.3394839

ACM Reference Format:
Michael Barz, Kristin Altmeyer, Sarah Malone, Luisa Lauer, and Daniel Sonntag. 2020. Digital Pen Features Predict Task Difficulty and User Performance of Cognitive Tests. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization (UMAP '20), July 14–17, 2020, Genoa, Italy*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3340631.3394839

1 INTRODUCTION

Digital pens enable the immediate digitization of handwritten texts and drawings into digital pen data which typically includes time-stamped spatial information and pressure grouped as pen strokes. Digital pens can be utilized for digitalizing analogue processes in education by, e.g., multimodal learning analytics [20], which enables technology-enhanced adaptive learning. Techniques that are developed for semantic annotation and interpretation of multimedia content [10, 12, 25, 26, 31, 39], automatic analysis of handwritten report forms [3] and gesture recognition for sketch-based interfaces [1, 2, 42] can be used for this purpose. In addition, signal-level features extracted from digital pens were shown to correlate with cognitive and affective states of a learner, e.g., expertise [21, 49], emotions [9, 36] and cognitive load [48, 50]. Hence, digital pens are a promising technology for scalable and real-time adaptive learning systems, also because recent advances in digital pen hardware for tablets[1] and paper[2] allow for an efficient and unobtrusive integration in existing learning environments.

In this work, we investigate the relation between pen-based features and task difficulty, as well as the learner's performance in a task. Other than the literature, we consider children as our target group. Measuring perceived task difficulty in children turned out to be challenging, as they might not have sufficient metacognitive skills to provide reliable self-report assessment [4, 5]. Therefore, behavioral and rather objective measures such as digital pen data seem to be advantageous for this target group. In contrast to retrospective measures, the online measures of the digital pen might provide much more detailed insights into children's task completion processes. Further, the digital pen fits into a child's familiar, natural work environment and thereby reduces extra mental load [19]. Due

[1] https://www.samsung.com/de/tablets/galaxy-tab-s6-t860/SM-T860NZAADBT/
[2] https://www.neosmartpen.com/en/

to the ongoing development of the prefrontal cortex and the related executive functions, children have difficulties controlling their attention and actions as well as inhibiting distracting impulses [6]. The non-intrusive measurement with a digital pen minimizes the distraction caused by data recording and does not interfere with task processing. The integrated sensors are not only more objective, they cover a different spectrum of observable features.

We focus on modelling task difficulty and user performance for two pen-based cognitive tests using support vector machines and gradient boosted trees. We include the Trail Making Test (TMT) for children [29, 30] and six drawing patterns from the Snijders-Oomen Non-verbal intelligence test (SON) [14]. Input features are extracted using a recently released feature library for digital pen data[3] that implements 165 features from the literature [24]. Prediction targets include the difficulty levels of the tasks (classification) and the continuous user performance measures for the TMT and SON tasks (regression): completion time for both tasks and the SON-specific measure pattern coverage. We systematically train and evaluate machine learning models using the nested crossvalidation paradigm and different approaches for feature selection. Also, we test the performance of our trained models on the respectively other task to estimate the generalizability of our models. The dataset for this work stems from a controlled lab study ($n = 36$): we asked children to solve two versions of the TMT and six drawing patterns of the SON test with increasing levels of difficulty. In addition, we implement and use the automatic metric *coverage* for SON patterns which encodes the proportion of the pattern that was solved correctly based on a distance threshold. The original evaluation scheme considers a manual and binary rating only.

Our main contributions are: (1) conducting a controlled experiment with children for collecting digital pen data from tasks with varying difficulty and (2) systematically modelling the relation of digital pen features to task difficulty and learner performance using combinations of two machine learning algorithms and generic feature selection approaches. Also, we discuss the impact of feature selection on the machine learning performance.

2 RELATED WORK

We describe related work on inference of users' cognitive states using digital pens, applications of digital pens in education, in particular for children in elementary school, and feature extraction methods for digital pens. We describe the background on the two cognitive tests used in this work, the TMT and SON tests.

2.1 Digital Pen Data and Cognitive States

Previous works investigated the relation between digital pen features and cognitive states of a user. Zhou et al. [49] investigated the performance of machine learning models for predicting domain expertise of a user and the dominant domain expert in a group of users for the Math Data Corpus [17]. They found that features based on average stroke distance, duration, pressure, and speed could effectively separate experts from non-experts in mathematics. A more detailed analysis of this experiment can be found in [21]. Frommel et al. [9] trained machine learning models that predict a user's affective state in a learning game using pen data from a digitizing tablet and in-game performance as input, and self-reported emotions as prediction target. Schrader and Kalyuga [36] determined that pen pressure can indicate emotions such as enjoyment and frustration. They state that the relation between pen pressure and writing performance is negotiated by the students' engagement.

Pen features can be used for diagnosis support in the medical domain [38]. Prange and Sonntag [27] used digital pens for automatic scoring of the clock drawing test for dementia diagnosis. An end-to-end approach for this was presented by Souillard-Mandar et al. [41]. Werner et al. [43] used signals from digital pens to differentiate between mild cognitive impairment and mild Alzheimer's disease. Other usecases include the diagnosis of Parkinson's disease [8], the improvement of the identification of children suffering from developmental disorders, e.g., coordination disorder [33], dysgraphia [32] or high-functioning autism spectrum disorder [34], by detecting deviant handwriting characteristics.

Other works focused on the relation of handwriting behavior and mental workload. Luria and Rosenblum [16] collected handwriting data for three tasks of increasing difficulty using a digitizing tablet (complete numerical progressions). They found differences in pen features based on temporal, spatial and angular velocity, but not on pressure, for the different levels of difficulty. Yu et al. [48] aimed at predicting the mental workload during sentence composition using curvature and velocity-based features. In previous research, they could also confirm an indicating function for pressure and pen orientation [46, 47]. Lin et al. [15] used three difficulty levels of an English sentence-making training to induce differences in cognitive load. By means of feature selecting techniques and machine learning, they successfully classified the resulting three levels of cognitive load through writing features. A cross-validation accuracy of 76.27% was reached by a subset of features including average pressure, azimuth, velocity in Y-direction, count of sensible pauses and maximum pressure. Zhou et al. [50] summarize further pen-based approaches for cognitive load estimation and describe the relation to the cognitive load theory and approaches based on other modalities. Similar to [16], we aim at differentiating the performed tasks using pen-based features under the assumption that the task difficulty has an observable effect on the handwriting behavior. However, we focus on cognitive tests that are performed by children and include more features. We concentrate on the sketching behavior which might differ from writing texts or digits [21, 45].

2.2 Digital Pens in Education

Digital pens can be applied to support students through interactive functions or to monitor their writing activity. An example of supportive interaction is the usage of the pen as assistive technology: Students with learning disabilities can benefit from digital pens providing an audio function [22]. Rawson et al. [28] investigated the use of the digital pen as monitoring device. They tracked student's homework activity using digital pen technology and found the productive use of homework time to be related to the course grade. Pen data can be used in multimodal learning analytics (MLA). The goal is to enable student-centered learning environments that support learning activities by modelling, predicting and reacting to (real-time) learning behavior and progress, e.g., cognitive load or domain expertise (see [18] for an overview).

[3] https://github.com/DFKI-Interactive-Machine-Learning/ink-features

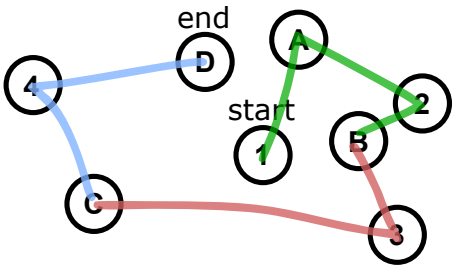


Figure 1: Shortened example pattern of the TMT-B task.

2.3 Feature Extraction from Digital Pen Data

Machine learning models require a representative encoding of sketches with respect to the prediction target, e.g., shapes or gestures. A common approach for signal analysis of digital pens is to devise a handcrafted set of geometric or temporal features including, e.g., stroke length, curvature and velocity. Different feature sets are proposed in the literature [7, 35, 40, 44] which were shown to be effective for training machine learning models. We integrate the comprehensive collection of hand-crafted features proposed in [24]. Alternatively, sketch representations can be learned from large-scale sketch datasets: Ha and Eck [11] presented the sketch dataset `QuickDraw` and the variational autoencoder `sketch-rnn` that learns to encode a sketch in a dense vector z for the purpose of reconstructing it. Kaiyrbekov and Sezgin [13] implemented a similar model, `stroke-rnn`, that encodes each stroke separately. They use the learned features (z vector) as input to a classification-based sketch segmentation model. However, both approaches remove timestamps and pressure data, which were shown to relate to cognitive states of a user [21, 36].

2.4 Cognitive Tests

The Trail Making Test for children is a standardized cognitive test that aims at measuring individual differences in the general executive function [29]. It includes two parts (A and B), each showing 15 encircled objects on a sheet of paper that shall be connected by drawing lines with a pen in the correct order and as quickly as possible. The performance indicator is the total completion time. `TMT-A` involves number sequencing (1 to 15), whereas `TMT-B` includes set-shifting: it requires the children to alternate between numerical and alphabetic sequences (1–A–2–B–3…). Completion times are expected to be higher in `TMT-B` since this condition places increased cognitive demands on children (see Figure 1 for an example).

The SON-R 5½-17 [37] is an adaptive, non-verbal intelligence test for children between five and a half and 17 years of age. It includes seven sub-tests in four categories: abstract reasoning tests, concrete reasoning tests, spatial tests, and perceptual tests. We use a set of items from the sub-test "patterns" which is a spatial test. It includes 18 repeating patterns consisting of one or two lines that correspond to 9 levels of difficulty. In the middle of each pattern, a small part is left out which has to be completed. It includes two items per level of difficulty, to allow for an adaptive testing procedure. The difficulty of a pattern results from the pattern's complexity (e.g., asymmetry, number of lines going backwards), the number of lines, and the width of the missing part (see Figure 2).

3 METHOD

In this section, we describe our approaches for modelling task difficulty and user performance using low-level features from digital pen data and our user study. We use classification for differentiating between multiple levels of difficulty and regression algorithms for estimating the task-related and continuous user performance metrics. In particular, we describe the feature extraction, the classification and regression problem, the feature selection approaches, and the details on model training. Further, we describe our user study and formulate research questions and corresponding hypotheses that guide our machine learning experiments.

3.1 Feature Extraction

The input to our machine learning models are digitized hand-drawn sketches S from digital pens. A single sketch $s \in S$ includes all pen strokes, i.e., time-stamped points in a 2-dimensional coordinate system and pressure values, that were drawn to solve one of the described tasks. For machine learning, sketches are commonly encoded using a set of hand-crafted features. We use of a recently published comprehensive collection of 165 features [24] that includes most of the previously proposed feature sets [7, 35, 40, 44]. Small feature subsets were already used for modelling cognitive states [21, 36, 49], but there is no explorative analysis that includes all features. We calculate the six additional features described in [49]: the average of the number of pen strokes, the average of the stroke distance and duration, the average of writing speed and pressure, and the total writing time. Finally, we represent each sketch s as a feature vector $f(s) \in \mathbb{R}^{171}$. For model training and inference, all sketches of a task S_{Task} are encoded into a feature matrix X_{Task} where each columns corresponds to one feature:

$$X_{Task} = f(S_{Task}) = \begin{bmatrix} f(s_1)^\top \\ \vdots \\ f(s_n)^\top \end{bmatrix}$$

Due to the high amount of available features, we implement and evaluate different feature selection strategies and discuss the resulting model performances. Including all features for model training might result in good prediction performance within a task, but is likely to overfit to the specific dataset. For all experiments, we remove three features that directly encode the completion time (one of our prediction targets) and one feature that, in our experiment, has zero variance[4]. 167 features remain for our experiments.

3.2 Classification of Task Difficulty

We consider task difficulty as a measure for mental effort that is required for solving a task: Our goal is to predict the task level and its inherent difficulty using data from the digital pen that was used for solving it. The *TMT* features two levels of difficulty, `TMT-A` (easy) and `TMT-B` (difficult). For *SON*, our tasks include six levels of difficulty, `SON-1` (easy) to `SON-6` (difficult). As we are interested in the generalizability of our models, we merge the patterns `SON-1` to `SON-3` (easy) and the patterns `SON-4` to `SON-6` (difficult). This allows an evaluation of our models across tasks and joint model training

[4] we exclude `rubine-13-duration`, `willems-24-duration` and the `average-writing-time` from [49] because they directly encode the completion time and `hbf49-36-2dhistogram` due to its zero variance.

by concatenating samples from both tasks. We hope to find features and weights that also estimate the difficulty of unseen pen-based tasks. The prediction target, *difficulty*, is defined per sketch s as:

$$d(s) = \begin{cases} 0 & \text{if } s \text{ is easy} \\ 1 & \text{if } s \text{ is difficult} \end{cases}$$

The vector y^d includes the difficulty levels of all sketches and is used for supervised learning and evaluation. We consider support vector classification using the radial basis function as kernel (SVC) and gradient boosted decision trees (GBDT) as machine learning algorithms. To merge the *SON* patterns, we extract the features separately for each sketch and use their element-wise mean. We report the performance of our classification models in terms of accuracy, the receiver operator characteristic (ROC) and the area under the ROC curve (AUC).

3.3 Estimating the User Performance

Similar to the classification setting, we consider the user performance metrics of both tasks as a measure for mental effort. The metrics include *time* for TMT and SON, as well as *coverage* for SON. The prediction target vectors used for supervised learning are y^{time} and $y^{coverage}$. We consider two regression algorithms for modelling the continuous metrics: support vector regression using the radial basis function as kernel (SVR) and a gradient boosted regression tree (GBRT). As the *time* metric is available for both tasks, we also consider a cross-task evaluation and joint modelling for estimating the generalizability of our models. We train separate regression models for predicting the *coverage* metric. The performance of our regression models are reported using the R^2 metric, i.e., the proportion of the variance in the prediction target that is explained by our features, and the mean squared error (MSE).

3.3.1 SON coverage. The assessment metric for SON is described in the administration manual [37]. It considers whether the reference pattern of the given task can be matched to the hand-drawn strokes or not. For this, the experimenter has to manually compare the drawn strokes with a reference pattern printed on a transparent foil. An item is solved, if the drawn pattern is complete and precise: the distance between a drawn stroke and the reference pattern may not exceed a maximum threshold defined by the diameter of a circle that is also printed on the foil. We implement an automatic assessment algorithm based on this principle. It takes the digitized sketch, the corner-points of the reference pattern in the same coordinate system and a distance threshold as input. We determine all parts of the reference model for which a pen signal exists that is closer than the distance threshold. We sum up the length of the parts of the reference pattern that are covered by the sketch. The ratio between this sum and the total length of the reference pattern (summed distances between the corner points) is our automatic metric *coverage*. The advantages of our metric are that it is much faster and more convenient to apply, and it is fully objective and more fine-grained than the manual evaluation. Figure 2 shows a SON-like example with overlaid digital pen data (original patterns may not be distributed). The reference pattern is visualized by their numbered corner points, green lines, if there is a successful match, and red lines, if there is no matching stroke data.

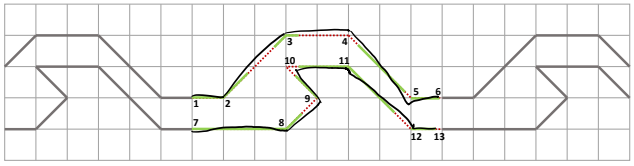


Figure 2: Example visualization of the SON coverage.

Table 1: Parameters used for hyper-parameter optimization by means of a grid search.

Model	Parameter Ranges
SVC and *SVR*	$C \in \{1, 10, 100, 1000\}$ $gamma \in \{10^{-3}, 10^{-4}\}$
GBDT and *GBRT*	$n\_estimators \in \{100, 500\}$ $max\_depth \in \{3, 4\}$ $learning\_rate \in \{.1, .01\}$

3.4 Feature Selection and Model Training

The goal of feature selection is to optimize the prediction performance of a model by selecting suitable features for a dataset. We consider two feature selection methods: the correlation-based method φ_{corr} and φ_{vif} based on the variance inflation factor. We compare them to a baseline that includes all features. We also test compositions of the two feature selection methods: $\varphi_{corr} \circ \varphi_{vif}$ and $\varphi_{vif} \circ \varphi_{corr}$. For φ_{corr}, we calculate the Pearson correlation between features and prediction target for TMT and SON, e.g., for φ^d_{corr}, we compute the correlation between X_{TMT} and y^d_{TMT}, and between X_{SON} and y^d_{SON}. The features with a correlation of $r > 0.2$ for both datasets are used for training. The VIF-based method φ_{vif} removes linear dependent features: features that cause multicollinearity in the training data X (independent from the prediciton target). We iteratively remove the features with the highest VIF score until no feature exceeds a threshold of 10.

All models are implemented as a scikit-learn [23] pipeline that applies two preprocessing steps: imputation of missing values using the mean value of a feature as replacement and robust data scaling (removes the median and scales the data based on the range between the first quartile and the third quartile). For training and evaluation, we use a 5x2 nested crossvalidation (CV): the inner loop performs a grid search (2-fold CV) for optimizing the model parameters, the outer loop (5-fold CV) estimates the generalization error of the trained model. The range of model parameters for the grid search is summarized in Table 1. To estimate how well a model generalizes across tasks, we consider two approaches: In case we train on samples from a single task (e.g., TMT), we compute the performance metrics for that model using samples from the respectively other task (e.g., SON). Alternatively, we can concatenate the task-specific datasets and regard the generalization error of the CV as generalization error. This requires that the datasets of both tasks include the same features and the prediction target has to be encoded equally. Concatenating the two datasets before training should improve the generalization capability of the model, because it can learn from samples of both tasks.

3.5 User Study

For our machine learning experiments, we use data from a controlled lab study: Elementary school children are provided with digital pens and instructed to perform two types of standardized cognitive tests (TMT and SON) involving drawing tasks. For both, we vary the difficulty of the sub-tasks and record the digital pen data and conventional performance measures (e.g., completion time). The labelled data is used in our machine learning experiments, i.e., for modelling the task difficulty and user performance based on digital pen data.

3.5.1 Participants. A total of 36 children (61% female) participated in this study. We invited students aged seven to eleven years (M = 10.04, SD = 1.3), because our target group are third and fourth grade elementary school children. The students attended the experimental setting in small groups of up to seven participants and were instructed simultaneously.

3.5.2 Study Design and Task Manipulation. In a within-subjects design, the factor task difficulty is varied in both cognitive tests. For the TMT, two difficulty levels are realized by providing each child with the `TMT-A` (easy) and `TMT-B` (difficult). For the sub-test "patterns" of SON, a set of six items, representing the first six of the nine levels of difficulty, is used. The children are instructed to complete all six items successively, beginning with the easiest one.

3.5.3 Procedure. First, the subjects receive information on the course of the experiment as well as on correct handling of the digital pen. Then, they are asked to note down their birth date, sex and handedness. After a standardized verbal task introduction, subjects solve the TMT as quickly as possible. Half of the participants start with `TMT-A`, the other half completes the `TMT-B` first. Each TMT part is preceded by visual demonstration and a sample trail. The subsequent SON-based task is explained by means of a sample pattern. Then, participants complete one of two randomly assigned parallel series of six incomplete patterns. The patterns of increasing difficulty are extracted and adapted from the SON-R 5½-17.

3.5.4 Apparatus. The study is conducted using seven instances of the Neo Smartpen M1 as digital pens. Each pen is connected to an Android mobile device via Bluetooth that runs a custom recording software based on the official Android SDK[5]. Our application visualizes the pen signal in real-time and streams it to a central recording server via local network or the internet (see Figure 3). The digital pen resembles a fountain pen in appearance, but is equipped with a standard ball pen tip, which facilitates writing on paper. The Neo Smartpen M1 uses an optical sensor for digitizing hand-drawn sketches including meta data such as the type of paper and the page. This requires that a subtle micro-dot pattern, Ncode, needs to be printed on the paper. We use the plain Ncode PDF-templates with page information for this study[6]. The digital pen provides strokes including high resolution pressure information. Based on the page and predefined bounding boxes for each task, we cluster the strokes into sketches that correspond to this task, e.g., the first SON pattern. Per participant, we store all raw data from the pen together with the task assignment using the human-readable JSON data format.

[5]https://github.com/NeoSmartpen/Android-SDK2.0
[6]https://www.neosmartpen.com/en/ncode-pdf/

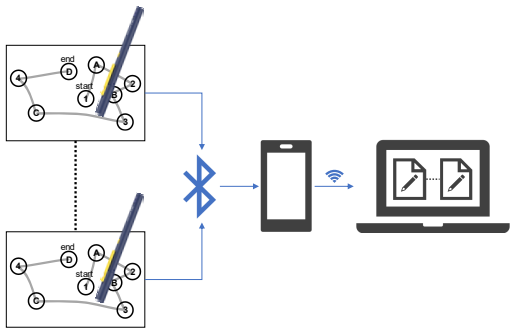


Figure 3: Technical architecture of the recording equipment.

3.5.5 Data Cleaning. We completely removed data from three participants, leaving data from 33 participants for our machine learning experiments. The data from participant 2 was removed, because of its age of 14 years (it is not part of our target group). Participant 4 showed no intention to solve the tasks correctly during the study: the participant was clearly scribbling for more than half of the tasks. The parents of participant 14 aborted the participation. Further, we observed that three participants either started too early or started within the bounding box of another task, both causing a long phase of inactivity in the beginning of individual sketches. This was possible due to the real-time visualization of the drawings that were observed by one of the experimenters. We manually removed the respective strokes in order to remove these inactivity phases. We excluded the data from participant 16 for the SON tasks, because the pen disconnected while recording them.

3.6 Machine Learning Experiments

We aim at modelling the difficulty d of a task and the user's performance, in terms of $time$ and $coverage$, in that task using features from a digital pen that was used for solving it. We investigate whether classification models can accurately differentiate between easy and difficult task levels of sketches $s \in S_{TMT}$ and $s \in S_{SON}$. We hypothesize that models perform well within a task in terms of accuracy (H1.1), but do not generalize well to other tasks because some features might be highly indicative for one of the tasks (H1.2). Further, we investigate the impact of the correlation-based feature selection φ^d_{corr}. The underlying assumptions are that (1) features that do not correlate with the prediction target y^d do not facilitate good predictions (garbage in, garbage out), and (2) features that correlate with the prediction target for only one of the tasks could cause overfitting for that task and, hence, reduce the generalizability. We expect that φ^d_{corr} has a positive effect on accuracy (H1.3). However, we are not sure about the effect on the generalizability between the two considered tasks, because the weights of the model are not influenced by samples of the respectively other task. Therefore, we consider model training and evaluation on concatenated datasets that include samples from both tasks: $S_{TMT+SON}$. Our hypothesis is that we can achieve a similar accuracy than for single task models and that the model benefits from φ^d_{corr} (H1.4).

For the regression models estimating the user performance metric $time$, we perform analogue tests to the classification problem

with similar hypotheses: We expect that regression models trained for one task can effectively predict the target *time* for that task (H2.1), but they do not generalize to the respectively unseen task (H2.2), and the φ_{corr}^{time} feature selection improves the performance of regression models in terms of the R^2 and *MSE* metrics (H2.3). In addition, we hypothesize that the completion time can be estimated well for both tasks, if $S_{TMT+SON}$ is used for training and that these models also benefits from the φ_{corr}^{time} feature selection (H2.4). Concerning S_{SON}, we also aim at predicting the metric *coverage*. Our hypothesis is that one of the considered models is able to learn the relation between $y^{coverage}$ and X_{SON} (H2.5).

For all classification and regression models, we investigate whether removing collinear features from the feature set improves the model performance: We apply the VIF-based feature filter to all generated datasets and repeat the model training. We also test whether the order application of the VIF-based and the correlation-based feature filter makes a difference. We assume that, to a certain degree, the model quality can be traded off against the number of features.

4 RESULTS

The results of the machine learning experiments for classification (predict the level of difficulty) are summarized in Table 2. We observe high accuracies for models that are trained and tested on samples from the same task using all features: The *SVC* achieves an accuracy of 83.19% for X_{TMT} and 90.99% for X_{SON}. The *GBDT* models perform better with accuracies of 100% and 97.03%, respectively. The φ_{corr}^{d} feature selection selects 52 features from the initial set of 167. On average, we observe a better accuracy for all four models, if the φ_{corr}^{d} feature selection is applied: the *SVC* accuracies increase around 8.46%, while the accuracies for the *GBDT* models slightly decrease by around 1.49%. However, all of these models perform poor for the respectively unseen task (around 50% or worse). If we perform the crossvalidation with data from both tasks ($X_{TMT+SON}$), we observe similar accuracy values compared to the single task models. Further, the φ_{corr}^{d} feature selection yields an improvement of 9.8% for *SVC* and a marginal improvement of 0.74% for *GBDT*. The φ_{vif} feature selection is applied to each dataset X separately and reduces the number of features to 25 for X_{TMT}, to 29 for X_{SON} and to 46 for $X_{TMT+SON}$. Overall, all accuracies are lower compared to their counterparts from the φ_{corr}^{d} feature selection. In a second run, we applied the φ_{vif} feature selection after φ_{corr}^{d} resulting in 11 features for X_{TMT} and X_{SON} and 13 features for $X_{TMT+SON}$. Compared to the φ_{vif}-based selection only, all accuracies are better. Despite the lower number of features that were used for training, the accuracies are close to the results from the φ_{corr}^{d}-based feature selection and even better in case of X_{TMT}. When applying φ_{corr}^{d} after φ_{vif} feature selection, the number of features can be reduced further (2, 3, and 7 features for X_{TMT}, X_{SON}, and $X_{TMT+SON}$, respectively), but at the cost of the model accuracies. Only the 7 remaining features for $X_{TMT+SON}$ yield a comparable accuracy of 92.39% with the *GBDT* model[7]. In Figure 4, we show the ROC curves and AUC scores of *SVR* and *GBDT* models trained using $X_{TMT+SON}$, $\varphi_{corr}^{d}(X_{TMT+SON})$ and $\varphi_{vif} \circ \varphi_{corr}^{d}(X_{TMT+SON})$ as datasets. The ROC curves and AUC scores indicate that the *GBDT* models have a better trade-off characteristic between true and false positive classifications. The *SVC* performs on par with the *GBDT* models for the $\varphi_{vif} \circ \varphi_{corr}^{d}$ feature selection. In addition, the two feature selection methods yield better characteristics than the baseline which includes all features. We observe the best ROC and AUC score for the *GBDT* model trained on $\varphi_{corr}^{d}(X_{TMT+SON})$, while the best accuracy score is observed for the corresponding *SVC* model.

The results of the regression experiments (predicting *time*) are summarized in Table 3. Using all features for training on X_{SON}, both models achieve good scores in the crossvalidation: R^2 is 0.71 for *SVR* and 0.7 for *GBRT*. For X_{TMT} the *SVR* model performs poor, while the *GBRT* model can explain some of the variance in y^{time} with $R^2 = 0.3$. The φ_{corr}^{time} feature selection selects 41 features, which has a positive impact on the model results. In particular, the *SVR* model trained on X_{TMT} achieves a better score of $R^2 = 0.43$, the *GBRT* model achieves a slightly better score of 0.46. For X_{SON}, the *SVR* model performs best with $R^2 = 0.88$. Analogue to our observation in classification, the task-specific models fail, if they are used with samples from the respectively unseen task: all R^2 scores fall below zero. The models for $X_{TMT+SON}$ achieve scores above zero: R^2 is 0.5 or 0.63 without feature selection and improves to 0.83 or 0.74, if φ_{corr}^{time} is used. We also investigate the impact of the φ_{vif} feature selection and its combinations with the φ_{corr}^{time} method. All models perform worse than the ones using φ_{corr}^{time} only. Only for the $\varphi_{vif} \circ \varphi_{corr}^{time}$ setting, i.e., applying the φ_{vif} after the φ_{corr}^{time} feature selection, the models perform well despite the further reduction of features: 13 features from X_{TMT} score to $R^2 = 0.44$ (*GBRT*) and 12 features from X_{SON}, as well as 16 features from $X_{TMT+SON}$ score to $R^2 = 0.77$ (*SVM*). A general trend is, that the models trained and evaluated on X_{SON} achieve better scores overall. Similarly, the *GBRT* models frequently perform better than their *SVR* counterparts, in particular, if the model needs to select from correlating and non-correlating features (i.e., if the φ_{corr}^{time} selection is not used). We also computed both scores for a mean baseline model, i.e., a model that always predicts the mean of the target variable. The *MSE* scores are 684.9 for X_{TMT}, 741.41 for X_{SON}, and 776.42 for $X_{TMT+SON}$. The R^2 score is zero by definition.

All regression models *coverage* as prediction target perform poor ($R^2 < 0$ for all models). We also tested a correlation-based feature selection, that selected features with mid and high correlation to $y^{coverage}$ from X_{SON} only, the φ_{vif} feature selection and combinations of them, but without success.

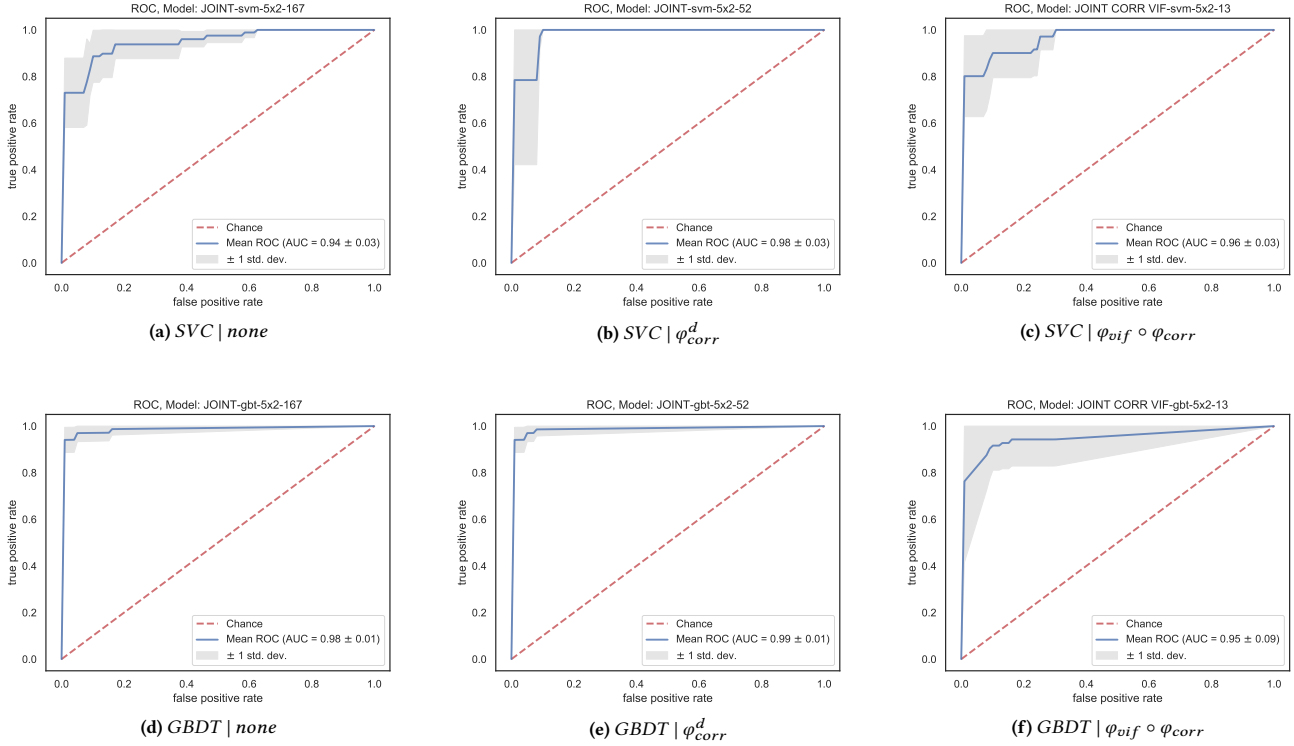
5 DISCUSSION

The results of our machine learning experiments indicate that we can distinguish well between easy and difficult task levels per cognitive test: we observe accuracies of 97.03% and 100% using the *GBDT* algorithm and all features (confirms H1.1). Testing the models with unseen samples from the respectively other task fails with accuracies around or below the chance level of 50% (confirms H1.2). Selecting features using φ_{corr}^{d} further improves the accuracy values within a task, but not across tasks. Even though the features correlate with the prediction target for all tasks, the learned weights seem to be test specific (we have to partially reject H1.3). We hypothesized that joint model training, i.e., including samples from both tasks for training, might solve this problem (H1.4). We can

[7] A table with the features selected by φ_{corr} and $\varphi_{vif} \circ \varphi_{corr}$ for classification and regression is provided in the appendix, because they performed well under all conditions.

Table 2: Accuracy for all classification models using different feature selection methods.

		Test Accuracy						
Feature Selection		*none*		φ^d_{corr}		φ_{vif}	$\varphi_{vif} \circ \varphi^d_{corr}$	$\varphi^d_{corr} \circ \varphi_{vif}$
X	**Model**	**5x2 CV**	**unseen**	**5x2 CV**	**unseen**	**5x2 CV**	**5x2 CV**	**5x2 CV**
X_{TMT}	*SVC*	83.19%	50.00%	93.96%	54.55%	87.69%	95.38%	75.93%
	GBDT	**100.00%**	51.20%	98.46%	30.91%	81.76%	95.38%	68.35%
X_{SON}	*SVC*	90.99%	50.00%	**97.14%**	47.58%	87.91%	92.53%	88.02%
	GBDT	97.03%	50.00%	95.60%	59.39%	93.96%	95.71%	87.91%
$X_{TMT+SON}$	*SVC*	87.21%	—	**97.01%**	—	87.15%	89.52%	88.60%
	GBDT	94.73%	—	95.47%	—	87.95%	93.25%	92.39%



(a) *SVC* | *none* **(b)** *SVC* | φ^d_{corr} **(c)** *SVC* | $\varphi_{vif} \circ \varphi_{corr}$

(d) *GBDT* | *none* **(e)** *GBDT* | φ^d_{corr} **(f)** *GBDT* | $\varphi_{vif} \circ \varphi_{corr}$

Figure 4: ROC curves and AUC scores for three *SVC* and three *GBDT* models using different feature selection strategies on $X_{TMT+SON}$. The models include 167 (*none*), 52 (φ^d_{corr}) or 13 ($\varphi_{vif} \circ \varphi_{corr}$) features.

Table 3: Model performance for all regression models and feature selection methods (R^2 / *MSE*).

		R^2 / *MSE* **test score**						
Feature Selection		*none*		φ^{time}_{corr}		φ_{vif}	$\varphi_{vif} \circ \varphi^{time}_{corr}$	$\varphi^{time}_{corr} \circ \varphi_{vif}$
X	**Model**	**5x2 CV**	**unseen**	**5x2 CV**	**unseen**	**5x2 CV**	**5x2 CV**	**5x2 CV**
X_{TMT}	*SVR*	-15.57 /> 10^4	-0.04 / 813.72	0.43 / 312.94	-0.83 / 1143.51	0.04 / 575.84	0.25 / 408.75	0.01 / 607.82
	GBRT	0.30 / 406.21	-1.51 / 1673.72	**0.46** / 374.28	-0.28 / 947.54	0.39 / 434.04	0.44 / **353.54**	-0.75 / 842.82
X_{SON}	*SVR*	0.71 / 220.40	-0.55 / 1064.26	**0.88** / **105.38**	-0.53 / 1048.43	0.63 / 238.44	0.77 / 179.20	0.08 / 673.63
	GBRT	0.70 / 194.79	-0.76 / 1711.40	0.76 / 194.79	-1.12 / 1711.40	0.77 / 152.36	0.74 / 231.34	0.17 / 600.89
$X_{TMT+SON}$	*SVR*	0.50 / 400.17	– / –	**0.83** / **123.50**	– / –	0.10 / 576.02	0.77 / 167.34	0.29 / 465.46
	GBRT	0.63 / 266.27	– / –	0.74 / 217.07	– / –	0.61 / 254.86	0.72 / 218.63	0.40 / 456.11

confirm it, because the models achieve accuracies of 87.21% (*SVR*) and 94.73% (*GBDT*) in predicting the correct level of difficulty and φ^{d}_{corr} yields a further improvement (only a marginal improvement for *GBDT*). This is also reflected in the ROC curves (see Figure 4): the models using φ^{d}_{corr} feature selection yield a better trade-off between true positive and false positive rates and the AUC scores are slightly better. Removing further features, e.g., using the VIF-based filter, results is worse model performances.

Another goal of our experiment is to show that the user performance for these tasks can be effectively modelled with regression models. The results show that our models can predict the *time* required for solving a task. This works well for X_{SON}, if all features are included, with an R^2 score of 0.71. For X_{TMT}, only the *GBRT* model can explain some of the variance in the target variable ($R^2 = 0.3$). This suggests that H2.1 can be confirmed. Analogue to the results from our classification experiments, hypotheses H2.2 and H2.3 for regression can be confirmed: the models do not generalize across tasks ($R^2 < 0$ for all models) and using φ^{time}_{corr} feature selection leads to improved model scores. Further, both regression algorithms can effectively learn to predict the completion time for both cognitive tests, if samples from both are present. The joint training particularly benefits from φ^{time}_{corr}, and similar to classification, the *SVR* algorithm benefits more (H2.4 can be confirmed). However, we have to reject H2.5, because all combinations of regression algorithms and feature selection methods with *coverage* as a prediction target yield poor performance metrics ($R^2 < 0$).

Overall, the results show that the φ_{corr} method effectively selects features from a larger set and improves the model performances for regression and classification. This is essential, if only a small sample size is available. In general, the gradient boosted tree models seems to be more robust to noise that is introduced by low correlating features: they show better performances, if all features are included. However, the support vector machine models benefit more from applying the additional feature selection (9.8% absolute improvement for classification and .33 absolute improvement for regression) which can lead to better performance than for gradient boosted tree models. From the remaining feature selection approaches, $\varphi_{vif} \circ \varphi_{corr}$ provides the best trade-off between number of selected features and model performance. We assume that features from φ_{corr}, that correlate with the prediction target, still suffer from multicollinearity. Removing collinear features using φ_{vif} from the remaining set of features does not deteriorate the model performances, because other, linear dependent features stay in the feature set and only little of the predictive power is lost.

5.1 Model Parameters.

We inspect the hyper-parameters of all classification and regression models using the φ_{corr} and the $\varphi_{vif} \circ \varphi_{corr}$ feature selection, because they tend to perform best across all tasks. Per machine learning algorithm, we merge the 5 best parameter sets from the nested crossvalidation from all three task datasets and report the most frequent parameter combination. These can be used as a starting point for future work. Regarding the gradient boosted tree models, the best parameter combination for classification (*GBDT*) is $learning\_rate = 0.1, max\_depth = 3, n\_estimators = 100$. For regression, the optimal value for $n\_estimators$ is typically 500. Regarding the models based on support vector machines, we observe $C = 100$ for classification and $C = 1000$ for regression. The most frequent value for *gamma* is 10^{-3}. Only for the φ_{corr}-based models, we observe an optimum of $gamma = 10^{-4}$.

5.2 Limitations & Future Work

We limit the scope of our discussion to the results in relation to our feature selection methods. It will be interesting to describe the selected features in detail for the classification of difficulty, estimation of completion time or both, as both can be seen as a measure for cognitive load. Selected and removed features should also be discussed with regard to features proposed in the literature. Another limitation is, that we merge the features of three easy and three difficult SON sketches two samples per participant for better comparison with the TMT samples. This might have an impact on model quality, because the difficulty increases gradually such that `SON-3` and `SON-4` have a similar difficulty, but belong to different classes. Further, it might be interesting to model the difficulty for SON as a six-class problem. It can also be beneficial for model training to normalize sketches before model training: scale sketches such that they fit into a quadratic box with a predefined edge length. Remaining challenges include the effective integration of pen-based user- and context models in technology-enhanced adaptive learning environments. We believe that digital pens are particularly suitable due to their unobtrusive nature. One possible application is to provide additional feedback about learning activities, e.g., homework, to teachers. This would enable more targeted and, hence, more efficient and effective interventions on an individual basis. In addition, a real-time estimation of task difficulty can be used to support learners by muting notifications of nearby devices.

6 CONCLUSION

We investigated whether features from digital pens can predict the difficulty of a task and the learner's performance in this task. We conducted a controlled user study to collect data and ran a systematic machine learning experiment with different learning algorithms and feature selection strategies. We could show that pen-based features can effectively predict the level of difficulty within a task and across tasks, if data from both tasks were used for training. The same holds for predicting the completion time as a performance measure. In addition, we implemented a fully automatic and more fine-grained evaluation algorithm for drawing patterns of the SON test. However, we could not predict this measure from digital pen data. The ability to precisely and unobtrusively estimate the task difficulty and user performance, which can be seen as measures for the cognitive load of a learner, opens up new opportunities for adaptive learning environments. This includes improvements in monitoring of the learning progress for more fine-grained feedback such as interventions from a teacher, but also automatic real-time adaption of digital learning environments in high load situations.

ACKNOWLEDGMENTS

This work was funded by the German Federal Ministry of Education and Research (BMBF) under grant number 16SV7768 (Intera-kt) and 01JD1811A, 01JD1811C (GeAR).

REFERENCES

[1] Lisa Anthony and Jacob O. Wobbrock. 2010. A lightweight multistroke recognizer for user interface prototypes. In *Proceedings of Graphics Interface 2010*. Canadian Information Processing Society, 245–252. https://dl.acm.org/citation.cfm?id=1839258

[2] Lisa Anthony and Jacob O. Wobbrock. 2012. $N-protractor: A Fast and Accurate Multistroke Recognizer. In *Proceedings of Graphics Interface 2012 (GI '12)*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 117–120. http://dl.acm.org/citation.cfm?id=2305276.2305296

[3] Michael Barz, Peter Poller, Martin Schneider, Sonja Zillner, and Daniel Sonntag. 2017. Human-in-the-Loop Control Processes in Gas Turbine Maintenance. In *Industrial Applications of Holonic and Multi-Agent Systems - 8th International Conference, HoloMAS 2017, Lyon, France, August 28-30, 2017, Proceedings (Lecture Notes in Computer Science)*, Vladimir Marik, Wolfgang Wahlster, Thomas I Strasser, and Petr Kadera (Eds.), Vol. 10444. Springer, 255–268. https://doi.org/10.1007/978-3-319-64635-0_19

[4] Natacha Borgers, Edith de Leeuw, and Joop Hox. 2000. Children as Respondents in Survey Research: Cognitive Development and Response Quality. *Bulletin of Sociological Methodology/Bulletin de Méthodologie Sociologique* 66, 1 (2000), 60–75. https://doi.org/10.1177/075910630006600106

[5] Christine Chambers and Charlotte Johnston. 2002. Developmental differences in children's use of rating scales. *Journal of Pediatric Psychology* 27, 1 (2002), 27–36. https://doi.org/10.1093/jpepsy/27.1.27

[6] Matthew C. Davidson, Dima Amso, Loren Cruess Anderson, and Adele Diamond. 2006. Development of cognitive control and executive functions from 4 to 13 years: Evidence from manipulations of memory, inhibition, and task switching. *Neuropsychologia* 44, 11 (2006), 2037–2078.

[7] Adrien Delaye and Eric Anquetil. 2013. HBF49 feature set: A first unified baseline for online symbol recognition. *Pattern Recognition* 46, 1 (2013), 117–130. https://doi.org/10.1016/j.patcog.2012.07.015

[8] Peter Drotár, Jiří Mekyska, Irena Rektorová, Lucia Masarová, Zdenek Smékal, and Marcos Faundez-Zanuy. 2014. Analysis of in-air movement in handwriting: A novel marker for Parkinson's disease. *Computer Methods and Programs in Biomedicine* 117, 3 (dec 2014), 405–411. https://doi.org/10.1016/J.CMPB.2014.08.007

[9] Julian Frommel, Claudia Schrader, and Michael Weber. 2018. Towards Emotion-based Adaptive Games: Emotion Recognition Via Input and Performance Features. In *The Annual Symposium on Computer-Human Interaction in Play Extended Abstracts - CHI PLAY '18*. ACM Press, New York, New York, USA, 173–185. https://doi.org/10.1145/3242671.3242672

[10] Achraf Ghorbel, Jean Camillerapp, and Aurélie Lemaitre. 2014. IMISketch: An interactive method for sketch recognition. *Pattern Recognition Letters* 35 (jan 2014), 78–90. https://doi.org/10.1016/J.PATREC.2013.08.011

[11] David Ha and Douglas Eck. 2017. A Neural Representation of Sketch Drawings. arXiv:cs.NE/1704.03477

[12] Tracy Hammond and Brandon Paulson. 2011. Recognizing sketched multistroke primitives. *ACM Transactions on Interactive Intelligent Systems* 1, 1 (oct 2011), 1–34. https://doi.org/10.1145/2030365.2030369

[13] Kurmanbek Kaiyrbekov and Metin Sezgin. 2019. Stroke-based sketched symbol reconstruction and segmentation. arXiv:cs.GR/1901.03427

[14] Jacob Arie Laros and Peter Johannes Tellegen. 1991. *Construction and validation of the SON-R 5 1/2-17, the Snijders-Oomen non-verbal intelligence test.* Ph.D. Dissertation. University of Groningen.

[15] Tao Lin, Tiantian Xie, Yu Chen, and Ningjiu Tang. 2013. Automatic cognitive load evaluation using writing features: An exploratory study. *International Journal of Industrial Ergonomics* 43, 3 (2013), 210–217. https://doi.org/10.1016/j.ergon.2013.02.002"

[16] Gil Luria and Sara Rosenblum. 2012. A computerized multidimensional measurement of mental workload via handwriting analysis. *Behavior Research Methods* 44, 2 (jun 2012), 575–586. https://doi.org/10.3758/s13428-011-0159-8

[17] Sharon Oviatt. 2013. Problem solving, domain expertise and learning: ground-truth performance results for math data corpus. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*. ACM, New York, NY, USA, 569–574. https://doi.org/10.1145/2522848.2533791

[18] Sharon Oviatt. 2018. Ten Opportunities and Challenges for Advancing Student-Centered Multimodal Learning Analytics. In *Proceedings of the 2018 on International Conference on Multimodal Interaction - ICMI '18*. ACM Press, New York, New York, USA, 87–94. https://doi.org/10.1145/3242969.3243010

[19] Sharon Oviatt, Alex Arthur, and Julia Cohen. 2006. Quiet Interfaces That Help Students Think. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST '06)*. Association for Computing Machinery, New York, NY, USA, 191–200. https://doi.org/10.1145/1166253.1166284

[20] Sharon Oviatt, Joseph Grafsgaard, Lei Chen, and Xavier Ochoa. 2018. Multimodal learning analytics: assessing learners' mental state during the process of learning. In *The Handbook of Multimodal-Multisensor Interfaces: Foundations, User Modeling, and Common Modality Combinations - Volume 2*. Association for Computing Machinery, 331–374. https://doi.org/10.1145/3107990.3108003

[21] Sharon Oviatt, Kevin Hang, Jianlong Zhou, Kun Yu, and Fang Chen. 2018. Dynamic Handwriting Signal Features Predict Domain Expertise. *ACM Transactions on Interactive Intelligent Systems* 8, 3 (jul 2018), 1–21. https://doi.org/10.1145/3213309

[22] Angela L. Patti and Krista Vince Garland. 2015. Smartpen Applications for Meeting the Needs of Students With Learning Disabilities in Inclusive Classrooms. *Journal of Special Education Technology* 30, 4 (2015), 238–244. https://doi.org/10.1177/0162643415623025 arXiv:https://doi.org/10.1177/0162643415623025

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

[24] Alexander Prange, Michael Barz, and Daniel Sonntag. 2018. A categorisation and implementation of digital pen features for behaviour characterisation. *CoRR* abs/1810.0 (oct 2018), 42. arXiv:1810.03970 http://arxiv.org/abs/1810.03970

[25] Alexander Prange, Danilo Schmidt, and Daniel Sonntag. 2017. A Digital Pen Based Tool for Instant Digitisation and Digitalisation of Biopsy Protocols. In *30th IEEE International Symposium on Computer-Based Medical Systems, CBMS 2017, Thessaloniki, Greece, June 22-24, 2017*, Panagiotis D Bamidis, Stathis Th. Konstantinidis, and Pedro Pereira Rodrigues (Eds.). IEEE Computer Society, 773–774. https://doi.org/10.1109/CBMS.2017.132

[26] Alexander Prange and Daniel Sonntag. 2016. Digital PI-RADS: Smartphone Sketches for Instant Knowledge Acquisition in Prostate Cancer Detection. In *29th IEEE International Symposium on Computer-Based Medical Systems, CBMS 2016, Belfast, UK and Dublin, Ireland, June 20-24, 2016*. IEEE Computer Society, 13–18. https://doi.org/10.1109/CBMS.2016.23

[27] Alexander Prange and Daniel Sonntag. 2019. Modeling Cognitive Status through Automatic Scoring of a Digital Version of the Clock Drawing Test. In *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization. International Conference on User Modeling, Adaptation, and Personalization (UMAP-2019), 27th, June 9-12, Larnaca, Cyprus*. ACM, New York, NY, USA, 70–77. https://doi.org/10.1145/3320435.3320452

[28] Kevin Rawson, Thomas F. Stahovich, and Richard Mayer. 2017. Homework and achievement: Using smartpen technology to find the connection. *Journal of Educational Psychology* 109, 2 (2017), 208–219. https://doi.org/10.1037/edu0000130

[29] Ralph M. Reitan. 1971. Trail Making Test Results for Normal and Brain-Damaged Children. *Perceptual and Motor Skills* 33, 2 (1971), 575–581. https://doi.org/10.2466/pms.1971.33.2.575 arXiv:https://doi.org/10.2466/pms.1971.33.2.575

[30] Ralph M Reitan. 1986. *Trail Making Test: Manual for administration and scoring*. Reitan Neuropsychology Laboratory.

[31] Hugo Romat, Nathalie Henry Riche, Ken Hinckley, Bongshin Lee, Caroline Appert, Emmanuel Pietriga, and Christopher Collins. 2019. ActiveInk: (Th)Inking with Data. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*. ACM Press, New York, New York, USA, 1–13. https://doi.org/10.1145/3290605.3300272

[32] Sara Rosenblum and Gideon Dror. 2016. Identifying Developmental Dysgraphia Characteristics Utilizing Handwriting Classification Methods. *IEEE Transactions on Human-Machine Systems* PP (12 2016), 1–7. https://doi.org/10.1109/THMS.2016.2628799

[33] Sara Rosenblum, Jumana Aassy Margieh, and Batya Engel-Yeger. 2013. Handwriting features of children with developmental coordination disorder - Results of triangular evaluation. *Research in Developmental Disabilities* 34, 11 (2013), 4134–4141. https://doi.org/10.1016/j.ridd.2013.08.009

[34] Sara Rosenblum, Hemda Simhon, and Eynat Gal. 2015. Unique handwriting performance characteristics of children with high-functioning autism spectrum disorder. *Research in Autism Spectrum Disorders* 23 (12 2015), 235–244. https://doi.org/10.1016/j.rasd.2015.11.004

[35] Dean Rubine. 1991. Specifying Gestures by Example. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '91)*. ACM, New York, NY, USA, 329–337. https://doi.org/10.1145/122718.122753

[36] Claudia Schrader and Slava Kalyuga. 2020. Linking students' emotions to engagement and writing performance when learning Japanese letters with a pen-based tablet: An investigation based on individual pen pressure parameters. *International Journal of Human Computer Studies* 135 (mar 2020), 102374. https://doi.org/10.1016/j.ijhcs.2019.102374

[37] Johannes T. Snijders, Peter J. Tellegen, and Jacob A. Laros. 2005. *Snijders-Oomen-Non-Verbal Intelligence Test.* (volume 3 ed.). Hogrefe, Göttingen. 244 pages.

[38] Daniel Sonntag. 2018. Interactive Cognitive Assessment Tools: A Case Study on Digital Pens for the Clinical Assessment of Dementia. *CoRR* abs/1810.0 (oct 2018), 7. arXiv:1810.04943 http://arxiv.org/abs/1810.04943

[39] Daniel Sonntag, Markus Weber, Alexander Cavallaro, and Matthias Hammon. 2014. Integrating Digital Pens in Breast Imaging for Instant Knowledge Acquisition. *AI Magazine* 35, 1 (mar 2014), 26. https://doi.org/10.1609/aimag.v35i1.2501

[40] Daniel Sonntag, Markus Weber, Alexander Cavallaro, and Matthias Hammon. 2014. Integrating Digital Pens in Breast Imaging for Instant Knowledge Acquisition. *AI Magazine* 35, 1 (2014), 26–37. http://www.aaai.org/ojs/index.php/aimagazine/article/view/2501

[41] William Souillard-Mandar, Randall Davis, Cynthia Rudin, Rhoda Au, David J Libon, Rodney Swenson, Catherine C Price, Melissa Lamar, and Dana L Penney. 2016. Learning classification models of cognitive conditions from subtle behaviors in the digital Clock Drawing Test. *Machine Learning* 102, 3 (mar 2016), 393–441. https://doi.org/10.1007/s10994-015-5529-5

[42] Radu-Daniel Vatavu, Lisa Anthony, and Jacob O. Wobbrock. 2012. Gestures as point clouds: a $P recognizer for user interface prototypes. In *Proceedings of the 14th ACM international conference on Multimodal interaction - ICMI '12*. ACM Press, New York, New York, USA, 273. https://doi.org/10.1145/2388676.2388732

[43] Perla Werner, Sara Rosenblum, Gady Bar-On, Jeremia Heinik, and Amos Korczyn. 2006. Handwriting Process Variables Discriminating Mild Alzheimer's Disease and Mild Cognitive Impairment. *The Journals of Gerontology: Series B* 61, 4 (07 2006), P228–P236. https://doi.org/10.1093/geronb/61.4.P228 arXiv:https://academic.oup.com/psychsocgerontology/article-pdf/61/4/P228/9909050/P228.pdf

[44] D.J.M. Willems and R. Niels. 2008. *Definitions for Features used in Online Pen Gesture Recognition*. Technical Report. NICI, Radboud University Nijmegen. http://unipen.nici.ru.nl/NicIcon/

[45] Kun Yu. 2016. Pen Input Based Measures. In *Robust Multimodal Cognitive Load Measurement*. Springer, 133–145.

[46] Kun Yu, Julien Epps, and Fang Chen. 2011. Cognitive Load Evaluation of Handwriting Using Stroke-Level Features. In *Proceedings of the 16th International Conference on Intelligent User Interfaces (IUI '11)*. Association for Computing Machinery, New York, NY, USA, 423–426. https://doi.org/10.1145/1943403.1943481

[47] Kun Yu, Julien Epps, and Fang Chen. 2011. Cognitive load measurement with pen orientation and pressure. In *Proceedings of MMCogEmS*. 4.

[48] Kun Yu, Julien Epps, and Fang Chen. 2013. Mental Workload Classification via Online Writing Features. In *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 1110–1114. https://doi.org/10.1109/ICDAR.2013.225

[49] Jianlong Zhou, Kevin Hang, Sharon Oviatt, Kun Yu, and Fang Chen. 2014. Combining empirical and machine learning techniques to predict math expertise using pen signal features. In *Proceedings of the 2014 ACM workshop on Multimodal Learning Analytics Workshop and Grand Challenge - MLA '14*. ACM Press, New York, New York, USA, 29–36. https://doi.org/10.1145/2666633.2666638

[50] Jianlong Zhou, Kun Yu, Fang Chen, Yang Wang, and Syed Z Arshad. 2018. Multimodal Behavioural and Physiological Signals as Indicators of Cognitive Load. In *The Handbook of Multimodal-Multisensor Interfaces: Foundations, User Modeling, and Common Modality Combinations - Volume 2*. Association for Computing Machinery and Morgan & Claypool, New York, NY, USA, Chapter X, 287–329.

Table 4: In the following table, we briefly summarize the features selected by the φ_{corr} and $\varphi_{vif} \circ \varphi_{corr}$ which performed well under all conditions.

		Selected Features
Classification (difficulty)	φ_{corr}^{d}	hbf49-01-side-ratio, hbf49-05-norm-first-to-last-vector, hbf49-06-cosine-flvector, hbf49-08-closure, hbf49-11-inflexion-x, hbf49-16-trajectory-length, hbf49-17-ratio-bb-length, hbf49-18-deviation, hbf49-19-avg-direction, hbf49-21-perpendicularity, hbf49-22-kperpendicularity, hbf49-24-dominant-direction, hbf49-25-dominant-direction, hbf49-26-dominant-direction, hbf49-27-dominant-direction, hbf49-28-local-changes-direction, hbf49-29-local-changes-direction, hbf49-32-2dhistogram, hbf49-33-2dhistogram, hbf49-38-2dhistogram, hbf49-43-hu-moment, hbf49-44-hu-moment, hbf49-49-compactness, markus-02-length, markus-03-area, markus-04-perimeter-length, markus-09-rectangularity, markus-10-closure, markus-11-curvature, markus-12-perpendicularity, rubine-05-distance-first-last-point, rubine-06-cosine-first-last-point, rubine-08-total-length, willems-01-trajectory-length, willems-11-pen-up-down-ratio, willems-12-average-direction, willems-13-perpendicularity, willems-14-average-perpendicularity, willems-15-deviation-perpendicularity, willems-25-average-velocity, willems-28-average-acceleration, willems-41-deviation-straight-line, willems-45-octant-sample-ratio, willems-47-octant-sample-ratio, willems-49-octant-sample-ratio, willems-59-distance-first-to-last, willems-62-absolute-curvature, willems-67-ratio-principal-axes, willems-68-average-centroidal-radius, willems-74-sin-chain-code, willems-78-cos-chain-code, willems-88-average-stroke-direction
	$\varphi_{vif} \circ \varphi_{corr}^{d}$ (TMT)	hbf49-11-inflexion-x, hbf49-44-hu-moment, hbf49-49-compactness, rubine-06-cosine-first-last-point, willems-11-pen-up-down-ratio, willems-28-average-acceleration, willems-49-octant-sample-ratio, willems-62-absolute-curvature, willems-74-sin-chain-code, willems-78-cos-chain-code, willems-88-average-stroke-direction
	$\varphi_{vif} \circ \varphi_{corr}^{d}$ (SON)	hbf49-11-inflexion-x, hbf49-22-kperpendicularity, hbf49-43-hu-moment, hbf49-49-compactness, rubine-06-cosine-first-last-point, willems-11-pen-up-down-ratio, willems-28-average-acceleration, willems-45-octant-sample-ratio, willems-74-sin-chain-code, willems-78-cos-chain-code, willems-88-average-stroke-direction
	$\varphi_{vif} \circ \varphi_{corr}^{d}$ (TMT+SON)	hbf49-11-inflexion-x, hbf49-18-deviation, hbf49-43-hu-moment, hbf49-49-compactness, rubine-06-cosine-first-last-point, willems-11-pen-up-down-ratio, willems-25-average-velocity, willems-28-average-acceleration, willems-49-octant-sample-ratio, willems-62-absolute-curvature, willems-74-sin-chain-code, willems-78-cos-chain-code, willems-88-average-stroke-direction
Regression (time)	φ_{corr}^{time}	avg-stroke-distance, avg-writing-speed, hbf49-01-side-ratio, hbf49-13-downstrokes-trajectory-proportion, hbf49-18-deviation, hbf49-19-avg-direction, hbf49-20-curvature, hbf49-21-perpendicularity, hbf49-24-dominant-direction, hbf49-25-dominant-direction, hbf49-26-dominant-direction, hbf49-28-local-changes-direction, hbf49-29-local-changes-direction, hbf49-33-2dhistogram, hbf49-34-2dhistogram, hbf49-35-2dhistogram, hbf49-38-2dhistogram, markus-11-curvature, markus-12-perpendicularity, rubine-09-total-angle-traversed, rubine-10-sum-of-absolute-angles, rubine-11-sum-of-squared-angles, willems-08-curvature, willems-11-pen-up-down-ratio', willems-12-average-direction, willems-13-perpendicularity, willems-14-average-perpendicularity, willems-15-deviation-perpendicularity, willems-16-centroid-offset, willems-21-maximum-angular-difference, willems-45-octant-sample-ratio, willems-47-octant-sample-ratio, willems-49-octant-sample-ratio, willems-50-octant-sample-ratio, willems-62-absolute-curvature, willems-72-sin-chain-code, willems-81-cos-chain-code, willems-83-cos-chain-code, willems-86-average-stroke-length, willems-87-standard-deviation-stroke-length, willems-88-average-stroke-direction
	$\varphi_{vif} \circ \varphi_{corr}^{time}$ (TMT)	avg-stroke-distance, avg-writing-speed, hbf49-13-downstrokes-trajectory-proportion, rubine-09-total-angle-traversed, rubine-11-sum-of-squared-angles, willems-11-pen-up-down-ratio, willems-16-centroid-offset, willems-49-octant-sample-ratio, willems-72-sin-chain-code, willems-81-cos-chain-code, willems-83-cos-chain-code, willems-87-standard-deviation-stroke-length, willems-88-average-stroke-direction
	$\varphi_{vif} \circ \varphi_{corr}^{time}$ (SON)	hbf49-13-downstrokes-trajectory-proportion, rubine-09-total-angle-traversed, rubine-11-sum-of-squared-angles, willems-11-pen-up-down-ratio, willems-16-centroid-offset, willems-49-octant-sample-ratio, willems-50-octant-sample-ratio, willems-72-sin-chain-code, willems-81-cos-chain-code, willems-83-cos-chain-code, willems-86-average-stroke-length, willems-87-standard-deviation-stroke-length
	$\varphi_{vif} \circ \varphi_{corr}^{time}$ (TMT+SON)	avg-stroke-distance, avg-writing-speed, hbf49-13-downstrokes-trajectory-proportion, hbf49-18-deviation, rubine-09-total-angle-traversed, rubine-11-sum-of-squared-angles, willems-11-pen-up-down-ratio, willems-16-centroid-offset, willems-45-octant-sample-ratio, willems-49-octant-sample-ratio, willems-50-octant-sample-ratio, willems-72-sin-chain-code, willems-81-cos-chain-code, willems-83-cos-chain-code, willems-87-standard-deviation-stroke-length, willems-88-average-stroke-direction