

# Unsupervised Anomaly Detection in Sequential Process Data

## Insights From PIAAC Problem-Solving Tasks

Okan Bulut<sup>1</sup> , Guher Gorgun<sup>2</sup> , and Surina He<sup>2</sup> 

<sup>1</sup>Centre for Research in Applied Measurement and Evaluation, University of Alberta, Canada

<sup>2</sup>Measurement, Evaluation, and Data Science, University of Alberta, Canada

**Abstract:** In this study, we present three types of unsupervised anomaly detection to identify anomalous test-takers based on their action sequences in problem-solving tasks. The first method relies on the use of the Isolation Forest algorithm to detect anomalous test-takers based on raw action sequences extracted from process data. The second method transforms raw action sequences into contextual embeddings using the Bidirectional Encoder Representations from Transformers (BERT) model and then applies the Isolation Forest algorithm to detect anomalous test-takers. The third method follows the same procedure as the second method, but it includes an intermediary step of dimensionality reduction for the contextual embeddings before applying the Isolation Forest algorithm for detecting anomalous cases. To compare the outcomes of the three methods, we analyze the log files from test-takers in the US sample ( $n = 2,021$ ) who completed the problem-solving in technology-rich environments (PSTRE) section of the Programme for the International Assessment of Adult Competencies (PIAAC) 2012 assessment. The results indicated that different groups of test-takers were flagged as anomalous depending on the representation (raw action sequences vs. contextual embeddings) and dimensionality of action sequences. Also, when the contextual embeddings were used, a larger number of test-takers were flagged by the Isolation Forest algorithm, indicating the sensitivity of this algorithm to the dimensionality of input data.

**Keywords:** action sequences, technology-rich items, PIAAC, anomaly detection, BERT, Isolation Forest

Process data refer to data stored in log files generated by individuals' interactions within a digital environment, such as computerized assessments in education (He et al., 2021). Records of events or actions occurring within a computer system or application can be used for various purposes, such as monitoring system performance, debugging software issues, auditing user activity, and detecting security threats. The structure of the log file is often similar across data sources or systems generating it, consisting of identifiers for users, timestamps, actions (or events), and action descriptions. Recorded actions in the log file mostly include keystroke operations (e.g., insert, delete, copy, paste, jump, and replace) and mouse clicks (e.g., click buttons, click links, double-click, select from dropdown menus, drag and drop, scrolling, zooming, searching, and sorting; Viswanathan & Vanlehn, 2017; Zhu et al., 2019).

In digital assessment environments, rich information stored in log files can be viewed as a supplementary source of information that extends far beyond mere response accuracy data (Goldhammer et al., 2017; Tang et al., 2021). Process data indicators extracted from log files can offer a profound glimpse into the intricate process of test-takers'

interactions while enabling more fine-grained analysis of test-takers' response process. Research over the past decade has extensively leveraged this resource for multifaceted purposes. For example, researchers have harnessed process data for identifying test-takers' problem-solving strategies (He & Davier, 2015; Stadler et al., 2019), generating more accurate scores at the group-level assessments (Shin et al., 2022), exploring different patterns of reading and writing behaviors (Hahnel et al., 2022; Zhu et al., 2019), and measuring test-takers' engagement and motivation levels (Nagy et al., 2022).

Recently, researchers have also begun to utilize sequential process data (e.g., action sequences in interactive problem-solving tasks) to examine test-takers' behavioral patterns using unsupervised learning approaches, such as *k*-means clustering (Hu et al., 2017), latent class analysis (Gao et al., 2022), and sequence mining techniques (He et al., 2022). For example, Xu et al. (2018) performed latent class analysis to categorize test-takers based on their action sequences in a complex problem-solving item in the 2012 Programme for International Student Assessment. The authors reported that the latent classes with higher intensities on certain types of actions (e.g., moving one

slider at a time in the item) had a higher probability of solving the item correctly than those taking inefficient actions frequently. He et al. (2019) employed the  $k$ -means algorithm to cluster test-takers based on their behavioral patterns in the Programme for the International Assessment of Adult Competencies (PIAAC) 2012 assessment. Using a single task from the problem-solving in technology-rich environments (PSTRE) section, He et al. found that the test-takers could be grouped into three clusters based on the number of actions where the clusters with a larger number of actions had better proficiency levels in the PSTRE tasks. Ullrich, He, and Pohl (2022) used sequential data mining to examine common behavioral patterns associated with incorrect responses in a PSTRE task from PIAAC 2012. Their findings showed that there were multiple groups of incorrect behavioral patterns due to differences in test-takers' level of effort and proficiency in subskills needed for solving the selected task correctly.

In this study, we aim to leverage another unsupervised learning approach, *anomaly detection*, to delineate distinct groups of test-takers by analyzing their behavioral patterns during interactive problem-solving tasks. Unsupervised anomaly detection is commonly employed to discern irregular or anomalous occurrences from regular or non-anomalous ones, as observed in atypical traffic flow within computer networks (Iglesias & Zseby, 2015) or fraudulent activity within credit card transactions (Rezapour, 2019). Unlike clustering and latent class analysis focusing on grouping observations with similar patterns, anomaly detection finds rare instances or outliers that deviate significantly from the norm. While this method has been adept at detecting aberrant response behaviors based on answer changes, hint requests, and response time (e.g., Gorgun & Bulut, 2022; Pan & Choe, 2021; Van der Linden & Jeon, 2012), it has not been utilized for exploring behavioral patterns within sequential process data. Thus, this study will expand the existing literature on sequential process data by delving into the application of unsupervised anomaly detection methods. The results of this study will offer new insights into how anomaly detection methods can discern irregularities within the sequences of test-takers' behaviors during problem-solving tasks, while also assessing the efficacy of these methods within the context of interactive problem-solving tasks.

In the remainder of this paper, we describe the extraction of information from sequential process data and discuss the use of anomaly detection methods for identifying anomalous behavioral patterns. Next, we describe three different methods to detect anomalous test-takers based on sequential process data, apply these methods to action sequences extracted from the PSTRE tasks in PIAAC 2012, and discuss the similarities and differences in

the results. Finally, we present the implications, limitations, and future directions of the study.

## Theoretical Framework

### Extracting Information From Process Data

A major challenge in extracting information from process data is data representation. Raw process data are often high-dimensional and unstructured, making them difficult to analyze and interpret (Y. Chen et al., 2022). For example, item U02 in PIAAC 2012 required test-takers to review a number of e-mails, identify relevant requests, and submit three meeting room requests using a simulated booking site. When solving this item, each action taken by test-takers is saved as a character string (e.g., START, FOLDER\_VIEWED, MAIL\_VIEWED\_1, REPLY, and SUBMIT\_UNFILLED) with a long-format structure in the log file. This leads to unstructured data with multiple entries for each test-taker corresponding to numerous actions performed on the task. The number of entries (i.e., actions) may vary from one test-taker to another because each test-taker is likely to follow a different approach to completing the task. Furthermore, many actions stored in the log file may not be relevant to the solution behavior (i.e., noisy actions), further complicating the analysis and interpretation of the process data (Tang et al., 2021).

To tackle the above challenges and extract as much information as possible from process data, researchers have proposed several information extraction methods, including  $n$ -gram language modeling (He & Davier, 2015), multidimensional scaling (MDS; Tang et al., 2020), and sequence-to-sequence autoencoders (Tang et al., 2021). The  $n$ -gram method is widely used in natural language processing (NLP) and computational linguistics to model continuous sequences of words, symbols, or tokens in a document. In the context of sequential process data,  $n$ -gram modeling treats an action sequence as a sequence of integers or symbols that can be broken down into  $n$ -grams (i.e., sequences of  $n$  adjacent symbols). After generating a large number of  $n$ -grams, feature selection methods can be performed to select the most informative ones representing important action sequences (He & Davier, 2015).

MDS is another technique to model high-dimensional process data involving action sequences (Tang et al., 2020). This exploratory method aims to create various latent variables (i.e., features) based on the dissimilarity of the action sequences and map them into Euclidean space with a user-defined number of dimensions. Dissimilarity measures such as the Levenshtein distance and Gómez-

Alonso and Valls's (2008) order-based sequence similarity are particularly suitable for vectors of action sequences with unequal lengths. Using raw features extracted from MDS, principal component analysis (PCA) can be performed to obtain orthogonal (i.e., uncorrelated) features with increased interpretability.

In addition to MDS, autoencoders have been used for dimension reduction and feature extraction with process data (Tang et al., 2021). Specifically, sequence-to-sequence encoders are applied to action sequences to reconstruct them with an encoder-decoder mechanism. The encoder converts the action sequences to low-dimensional vectors (i.e., action embeddings), while the decoder attempts to generate an output similar to the original input using a recurrent neural network. As for MDS, raw features obtained from the autoencoder can be transformed into orthogonal features with principal component analysis. Although the sequence-to-sequence autoencoder method produces low-dimensional latent vectors representing the original action sequences, it can only process the input sequences in left-to-right or right-to-left order (i.e., one direction), limiting its ability to capture long-term dependencies in the action sequences.

## Sequential Process Data as Contextual Embeddings

Recent studies have shown that sequential process data can be interpreted as a natural language sequence (e.g., Landauer et al., 2023), enabling the numerical representation of such data as semantic vectors (i.e., embeddings). In NLP, an embedding is essentially a numerical representation of a categorical feature (e.g., a word) with floating point values. Since each action sequence resembles a sentence comprising multiple words, a semantic model can be used to obtain embeddings for the unique actions (i.e., tokens) stored in the sequence. For example, a pretrained transformer, such as Bidirectional Encoder Representations from Transformers (BERT; Devlin et al., 2018), can use semantic encoding to learn contextual relations between the actions within a sequence and produce an embedding for each sequence. Not only does this process put the actions (i.e., character strings) into a numerical vector space, but it also maintains the relative positions of the actions in the sequence.

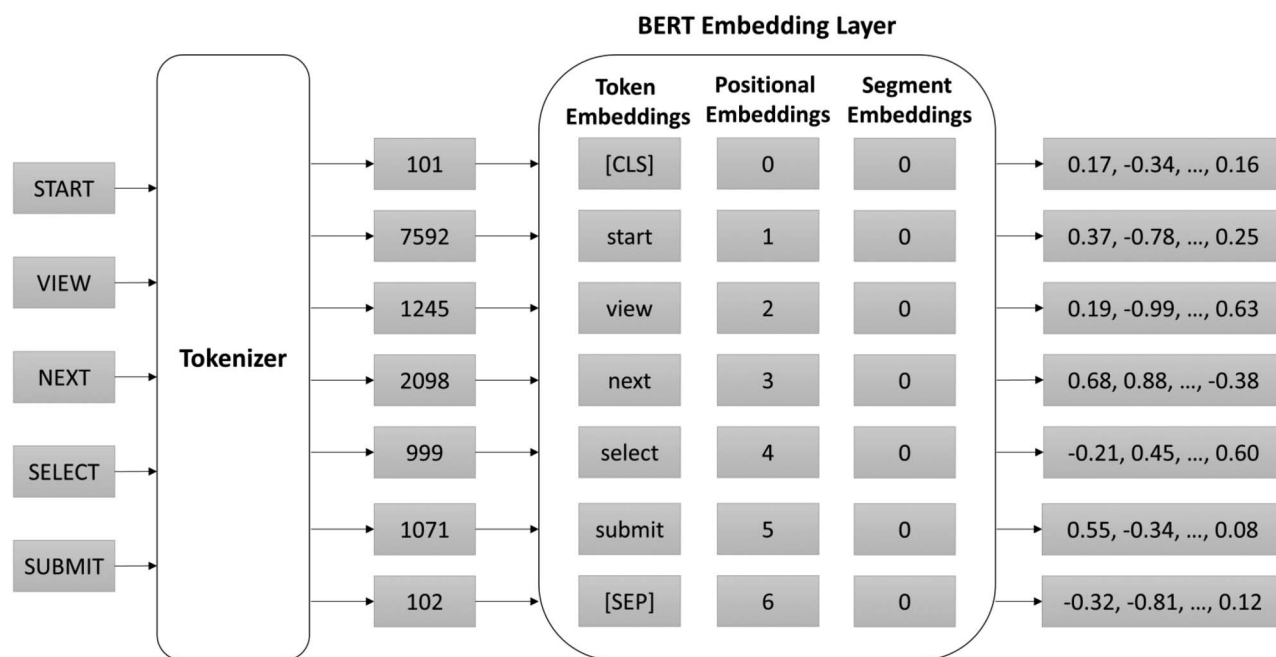
Pretrained transformers, including BERT, can convert a sequence into embeddings in multiple steps. First, the input sequence (e.g., a sentence) is transformed into individual tokens (e.g., words or subwords) where each token is assigned a unique numerical identifier (token ID) from the model's vocabulary. Each token is then represented as an initial embedding in the model. Next, positional encodings

are generated for the initial embeddings to capture information about the position of each token in the sequence. Transformers consist of multiple layers, each containing two main sublayers (i.e., the self-attention mechanism and a feedforward neural network). The self-attention mechanism is used to place the embeddings onto a vector space, where similar instances are positioned closer to each other than less similar ones (Guo et al., 2021; Landauer et al., 2023). This process assigns weights (i.e., attention scores) to specific inputs based on the relationships between all pairs of tokens in the sequence. The output from the self-attention mechanism is passed through a feedforward neural network within the same layer. This process is repeated for each layer in the transformer via layer stacking, making the output of one layer serve as the input to the next layer. At the end, the output embeddings from the final layer are pooled to obtain a fixed-size representation of the entire input sequence.

Figure 1 illustrates how BERT, as a pretrained transformer, can transform a hypothetical action sequence, ["START," "VIEW," "NEXT," "SELECT," "SUBMIT"], into embeddings. BERT has three distinct types of embeddings: token embeddings, positional embeddings, and segment embeddings to represent sequential input data. As described above, token embeddings refer to a list of token IDs based on unique actions in the sequence, while positional embeddings show token position within the sequence. Segment embeddings are a list of IDs distinguishing different sequences. Segment embeddings may not be relevant in this example unless the action sequence is divided into multiple action blocks (i.e., segments). To indicate the beginning and end of the sequence, the BERT Tokenizer adds two new tokens, [CLS] and [SEP], to the sequence and then assigns a unique ID to each token from its vocabulary containing all English characters and the 30,522 common words and subwords found in the corpus the model was trained on. Next, it creates token, positional, and segment embeddings for the tokens within the embedding layer. In the final step, the BERT embedding layer is used to compute the final embedding with 768 values for each input token by combining the token, positional, and segment embeddings (see Devlin et al., 2018, for further details on the model architecture). Since BERT can analyze the action sequences in both forward and backward directions, it can capture long-term dependencies and discrepancies in the problem-solving process.

## Anomaly Detection With Process Data

The information obtained from log files has been used for addressing a wide range of research questions, such as



**Figure 1.** Transforming action sequences into contextual embeddings via BERT.

identifying problem-solving strategies (Stadler et al., 2019), exploring different patterns of behaviors (Hahnel et al., 2022; Zhu et al., 2019), and measuring test-takers' ability, engagement, and motivation levels (Nagy et al., 2022; Xiao et al., 2021). However, only a few studies have utilized process data for anomaly detection (e.g., Gorgun & Bulut, 2022; Liao et al., 2021). The term *anomaly* (also referred to as aberrance or outlier in the literature) describes rare events, items, or observations that differ significantly from standard (i.e., norm) patterns. In educational testing, anomalous cases (e.g., responses or test scores) are often identified at individual test-taker and group levels based on aberrant response behaviors, such as careless responding (Ulitzsch, Yildirim-Erbasli, et al., 2022), unmotivated responding (Johns & Woolf, 2006), hint abuse or overuse (Gorgun & Bulut, 2022), and cheating (Kamalov et al., 2021; Kim et al., 2016). However, anomalous cases may also stem from a positive solution behavior. For example, when undertaking a complex task, test-takers may have to identify a set of manageable subgoals and perform corresponding actions (He et al., 2021). In this process, a creative strategy for breaking down the tasks very efficiently could be considered a desirable form of anomaly.

Anomaly detection methods for log files can be categorized into two groups based on the availability of labels for identifying usual and anomalous cases (Landauer et al., 2023; Meena Siwach & Mann, 2022). Supervised machine learning (ML) methods, such as logistic regression, support

vector machines, and decision trees, are applied to labeled data to learn the difference between usual and anomalous cases based on a set of predictors (Omar et al., 2013). However, anomaly detection often boils down to the problem of finding anomalous cases via unsupervised ML methods without ground truth (i.e., pre-existing labels for regular and anomalous cases). This is particularly challenging when dealing with large amounts of unstructured log files for which manual labeling by human annotators may not be feasible. Thus, unsupervised anomaly detection methods (e.g., Isolation Forest, Local Outlier Factor, Elliptic Envelope, and Log Clustering) are often used to establish a profile of usual data points and report anomalous cases deviating from this profile. In a recent study, Gorgun and Bulut (2022) used several unsupervised methods (e.g., Gaussian Mixture Model, Isolation Forest, and Local Outlier Factor) to identify aberrant responses in the intelligent tutoring system based on response time, action frequency, and response accuracy. Their findings showed that the unsupervised methods flagged similar responses as aberrant, which had negative correlations with students' level of concentration and positive correlations with their boredom.

In recent years, various deep learning methods have been proposed for anomaly detection in the system log files generated by an operating system, application, server, and so on (e.g., Catillo et al., 2022; Le & Zhang, 2021; Wittkopp et al., 2021). Furthermore, some researchers have combined these methods with NLP techniques to

facilitate the processing of complex messages in log files (e.g., S. Chen & Liao, 2022; Guo et al., 2021; Ryciak et al., 2022; Shao et al., 2022). With pretrained large language models such as BERT (Devlin et al., 2018), sequential log entries over time or across multiple users are transformed into dense, numerical vectors (i.e., embeddings) that retain the order and dependency of the entries in a lower dimensional space. Then, a deep learning algorithm for labeled data (e.g., recurrent neural networks) or a clustering algorithm for unlabeled data (e.g., *k*-means clustering) can be applied to the embeddings for identifying clusters with usual log sequences and finding other clusters with significant deviations from the usual log sequences (e.g., anomalous cases). Previous research suggests that NLP-based methods can improve the accuracy of anomaly detection and make further analysis of suspicious sequences much easier (Ryciak et al., 2022).

To date, several studies have combined NLP-based methods with unsupervised ML techniques to detect anomalies using log files from computer and network systems. For example, Wang and AnilKumar (2023) used linguistic models, such as BERT and Word2Vec, to extract features (i.e., embeddings) from HTTP traffic packets, performed PCA and autoencoders to reduce the dimensionality of embeddings, and then used the lower-dimensional embeddings to detect system anomalies based on unsupervised techniques (e.g., One-Class Support Vector Machine, Isolation Forest, and Local Outlier Factor). In a recent study, Karlsen (2023) also used several pretrained transformers (e.g., BERT, RoBERTa, GPT-2, and GPT-NEO) to extract contextual embeddings from system logs files. After the feature extraction procedure, they used several unsupervised methods (e.g., *k*-means, hierarchical clustering, Isolation Forest, and Self-Organizing Maps) to cluster the features and detect anomalies. The findings of Karlsen's study showed that the large language models exhibited enhanced capabilities in anomaly detection by extracting meaningful contextual embeddings.

Despite the advancements in leveraging NLP-based methods for anomaly detection in computer and network systems, a notable gap remains in the application of feature extraction with language models to sequential process data. While existing studies have successfully employed linguistic models such as BERT, Word2Vec, and various pretrained transformers to extract features from log files and enhance anomaly detection in system-related activities, there is a paucity of research focusing on the application of these techniques to sequential processes. Sequential process data with a time-ordered sequence of events (e.g., test-takers' actions in interactive problem-solving tasks) pose unique challenges that require specialized approaches. Addressing this gap could significantly contribute to improving the detection of anomalies in

dynamic and evolving systems where the temporal aspect plays a crucial role. Thus, further research is needed to explore and adapt language model-based feature extraction methodologies to the specific characteristics of sequential process data for building an effective anomaly detection framework.

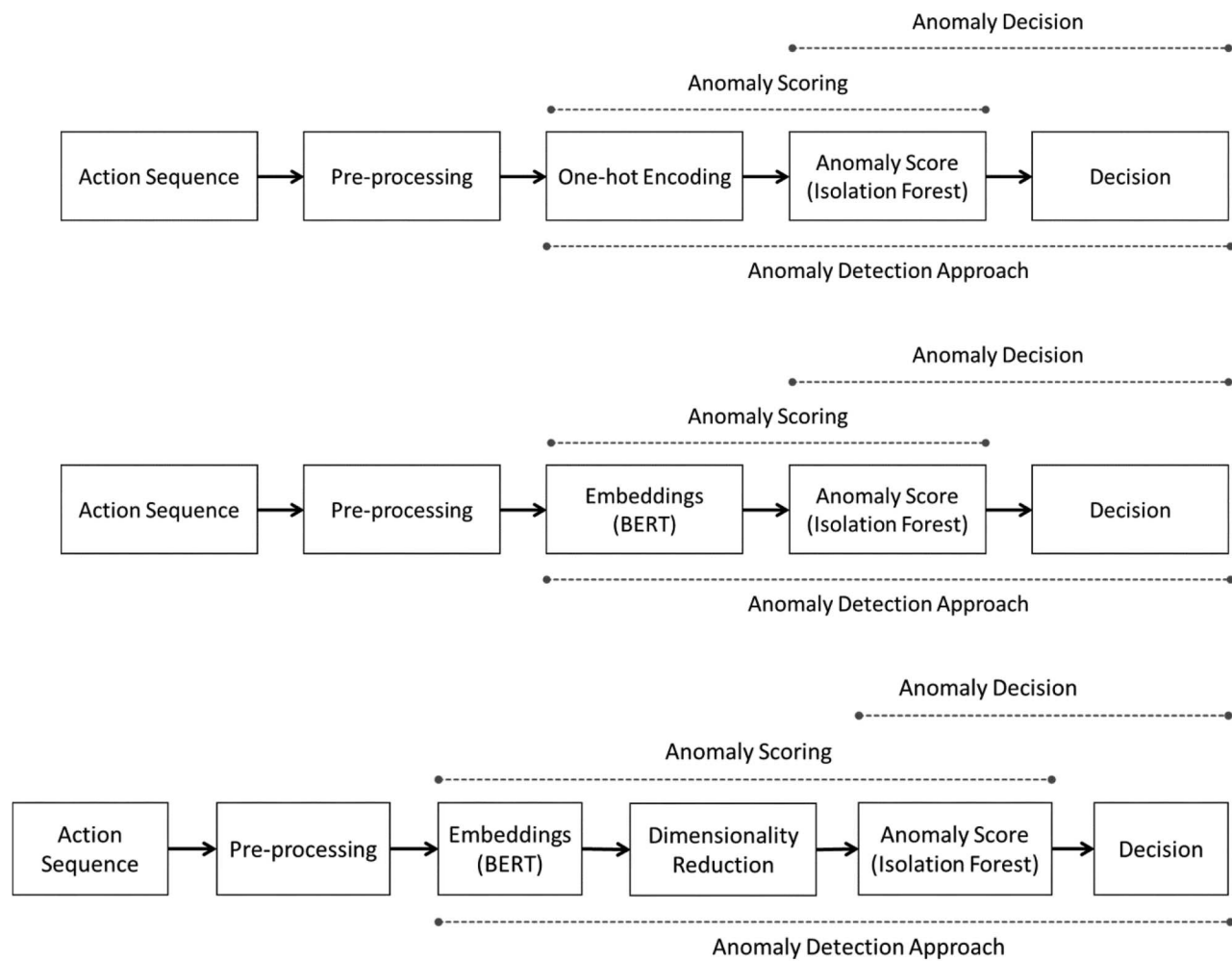
## Current Study

This study aimed to investigate whether unsupervised anomaly detection could help distinguish regular and anomalous test-takers based on their action sequences in the PSTRE domain of PIAAC 2012. Using the log files from American adults who participated in PIAAC 2012, we employed three unsupervised anomaly detection methods. In the first method, we applied the Isolation Forest algorithm to detect anomalous test-takers based on raw action sequences extracted from the log files. In the second method, we first transformed the raw action sequences into contextual embeddings using BERT and then performed unsupervised anomaly detection using Isolation Forest. The third method also started with the transformation of raw action sequences into contextual embeddings using BERT. However, as an intermediary step, PCA was performed to put the embeddings onto a lower-dimensional space. In the final stage, the Isolation Forest algorithm was applied to the lower-dimensional data to detect anomalous test-takers (see Figure 2 for an overview of the anomaly detection methods). To compare the three methods, we examined several test-related outcomes (e.g., average sequence length, response time, and incorrect response rates) and the distribution of PSTRE proficiency levels for anomalous and regular test-takers. Also, we explored the relationship between anomaly status and test-taker characteristics (e.g., informational and communicative technology [ICT] use at home and work) associated with test-takers' problem-solving behaviors in PIAAC 2012 (Liao et al., 2019; Zhang et al., 2021).

## Methods

### Sample and Instruments

Data for the current study came from the 2012 administration of PIAAC – an initiative of the Organisation for Economic Co-operation and Development (OECD) to assess the proficiency levels of adults between the ages of 16 and 65 in several information-processing skills (e.g., literacy, numeracy, and problem-solving). This study focused on the PSTRE domain of PIAAC with 14 items (seven items



**Figure 2.** Anomaly detection methods for sequential process data. Adapted from Wittkopp et al. (2021).

in each of the two booklets) measuring the intersection of computer literacy and the cognitive skills required to solve digital problem-solving tasks (Organization for Economic Cooperation and Development [OECD], 2019).<sup>1</sup> We selected the PSTRE domain because, in addition to the response accuracy data with correct, incorrect, and partially correct responses, there is also a log file containing process data on the participants' behavioral patterns (i.e., action sequences and the associated times) for the PSTRE items. As the participants interacted with the items in different environments (i.e., e-mail, web, word processor, and spreadsheet), log files recorded the actions taken during the assessment, such as opening a folder, clicking a link, and using the help function.

We used the participants in the US sample ( $n = 2,021$ ) who completed the PSTRE items in the first ( $n = 1,341$ ) or second

( $n = 680$ ) booklet and the background questionnaire in PIAAC 2012. In the final sample, there were 1,090 female participants (54%) and 931 male participants (46%). There were nearly equal proportions ( $\sim 20\%$ ) of participants from each age group (24 or less, 25–34, 35–44, 45–54, and 55+). Almost half of the sample (49%) reported their highest level of schooling as either high school or less than high school, while the other half had an educational level above high school (51%). Furthermore, most of the participants (89%) were native speakers of English. Finally, 76% of the participants were employed, and the remaining 24% were unemployed or out of the labor force when participating in PIAAC 2012. Also, the PIAAC 2012 database included additional information on the participants, such as their engagement in literacy and numeracy activities, use of ICT at home and work, occupation, and immigration status.

<sup>1</sup> The PSTRE items are not publicly available, but sample PSTRE items in PIAAC 2021 can be seen at <http://www.oecd.org/skills/piaac/Problem%20Solving%20in%20TRE%20Sample%20Items.pdf>.

## Extracting Process Data

Sequential process data for the present study were extracted using several steps. First, we used the LogDataAnalyzer software program on the PIAAC log file website to transform the raw log file in the .xml format into a structured dataset in a long format (i.e., multiple rows for each participant corresponding to actions taken to solve each PSTRE item). Second, we replaced the underscore separating some action descriptions with a space (e.g., “FOLDER\_VIEWED” to “FOLDER VIEWED” in item U01A – Party Invitations) to facilitate the tokenization process. Third, to create a single line of action sequences, we reshaped the dataset to a wide format where each participant had a vector of action sequences separated by a hyphen (e.g., [START - FOLDER VIEWED - ... - NEXT ITEM - END]). In the last step, we changed the vector of action sequences from uppercase to lowercase. These data extraction steps were completed using the R programming language (R Core Team, 2022).

Table 1 presents a descriptive summary of action sequences and proportion-correct scores for the PSTRE items. Although the minimum length of action sequences was similar across the items (ranging from 3 to 10), the maximum length varied significantly (ranging from 68 to 1,179), indicating a distinctive problem-solving process underlying each PSTRE item. The number of unique actions for each item indicated a moderate relationship ( $r = 0.45$ ) with the average sequence length because the test-takers had to use some actions repeatedly as they completed the task. For example, U04A – Class Attendance with relatively fewer unique actions (25) had the longest

average sequence (122.49) among all PSTRE items. As the test-takers solved this item, they had to repeatedly use DOACTION (i.e., a user interaction related to triggering a programmatic function; 62,654 times). Furthermore, item difficulty (i.e., average scores) indicated a negative relationship ( $r = -0.36$ ) with the average length of action sequences, suggesting that the difficult items (e.g., U04A – Class Attendance and U02 – Meeting Room) generally involved longer action sequences than the easier items in the PSTRE test. In addition, the items in the second booklet (i.e., U19A to U23) were generally more straightforward and required more actions on average than those presented in the first booklet (i.e., U01A to U04A).

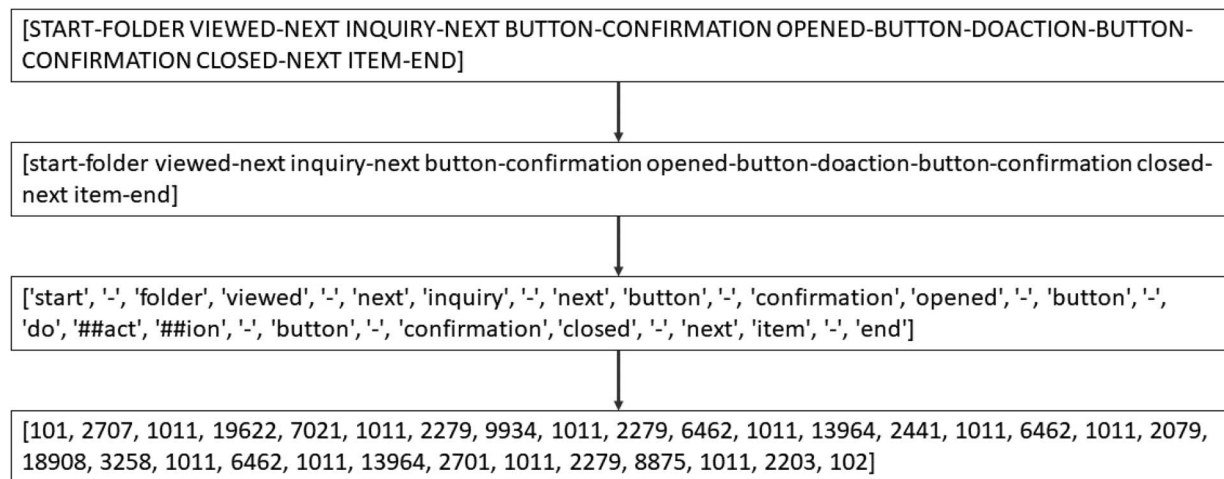
## Computing Contextual Embeddings

To obtain contextual embeddings for the action sequences, we first tokenized the action sequences for each item using the tokenizer function from the pretrained BERT base model, including 12 encoder layers with 768 hidden units and 110 million trainable parameters (i.e., bert-base-uncased; Devlin et al., 2018). The tokenizer checked if each word in the action descriptions was included in BERT’s vocabulary. Words that could not be found in BERT’s vocabulary were broken into the largest possible word contained in the vocabulary or decomposed into individual characters (e.g., “action” became “##act” and “##ion”). This process yielded a vector of up to 512 tokens, which is the maximum length of input sequence for BERT. Padding was applied to ensure shorter action

**Table 1.** Descriptive statistics for action sequences (Seq.) in PSTRE items

Booklet	Item ID	Task name	Min seq. length	Max seq. length	<i>M</i> ( <i>SD</i> ) seq. length	Average score (%)	Number of unique actions
1	U01A	Party invitations	3	223	38.29 (23.19)	61.53	37
1	U01B	Party invitations	7	520	68.09 (47.92)	43.33	38
1	U03A	CD Tally	8	371	36.47 (33.64)	36.76	26
1	U06A	Sprained ankle	3	86	19.15 (7.77)	26.69	18
1	U06B	Sprained ankle	10	119	36.98 (20.36)	49.74	28
1	U21	Tickets	10	153	40.87 (16.16)	38.60	24
1	U04A	Class attendance	10	857	122.49 (139.96)	13.04	25
2	U19A	Club membership	10	268	48.03 (27.23)	65.82	26
2	U19B	Club membership	10	349	53.73 (41.94)	55.22	21
2	U07	Book order	10	1,047	46.44 (39.40)	46.49	24
2	U02	Meeting room	10	1,179	101.96 (120.14)	19.60	56
2	U16	Reply all	4	822	105.16 (103.48)	52.95	36
2	U11B	Locate email	10	505	42.80 (40.83)	29.00	35
2	U23	Lamp return	3	637	51.27 (48.42)	37.32	38

Note. The order of PSTRE items reflects the actual item positions in PIAAC 2012. The first seven PSTRE items were administered in the first booklet, whereas the last seven PSTRE items were administered in the second booklet.



**Figure 3.** Transforming PSTRE action sequences into token indices (Item U01A – Party Invitations).

sequences were the same length (i.e., the maximum length of 512 tokens), resulting in vectors ending with zeroes for participants with less than 512 actions for a given item. Also, the tokenizer captured the positional embeddings for each token to show the token position within the action sequence and the segment embeddings representing each action sequence. After breaking the action sequences into tokens, we converted the vector of tokens to a vector of token indices (i.e., a unique identification number for each token). Figure 3 shows how a test-taker's actions for U01A – Party Invitations were transformed into a vector of token indices. In the final step, we ran the three vectors (token, positional, and segment embeddings) through the BERT embedding layer to form a 768-dimensional embedding vector based on the last layer (S. Chen & Liao, 2022).

## Anomaly Detection With Isolation Forest

To detect anomalous cases in the action sequences, we used the Isolation Forest (also referred to as iForest) algorithm. Previous studies indicated that Isolation Forest is a robust algorithm capable of detecting anomalous cases in high-dimensional data (Gorgun & Bulut, 2022; Liu et al., 2008). Unlike distance-, density-, or model-based methods prioritizing the construction of a profile of regular instances to detect anomalies, Isolation Forest aims to isolate instances of unusual instances using a tree structure (Liu et al., 2008). The central assumption underlying anomaly detection with Isolation Forest is that unusual instances are different and less frequent than regular instances, and thus, they can be isolated quickly in the partitioning process, whereas regular instances are likely to require more partitions in tree-based structures.

Isolation Forest starts by randomly selecting a point between the maximum and minimum values of a feature randomly picked from the dataset. Then, the selected feature's range is used to split the tree into two nodes based on the randomly chosen value (smaller values on the left and larger values on the right). This iterative process continues until all the instances in the dataset are partitioned. Then, shorter paths in the tree structure are flagged as unexpected instances (i.e., anomalies). The algorithm can easily distinguish those instances from the regular instances because the flagged instances have shorter paths than regular instances. An anomaly score between 0 and 1 is calculated for each instance by comparing its path length to the expected path length for regular instances, and scores larger than 0.5 (or a user-defined threshold) are considered anomalous. Mathematically, this can be written as follows:

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}}, \quad (1)$$

where  $h(x)$  represents the length of the path of observation  $x$ ,  $E(h(x))$  is the average  $h(x)$  from a collection of isolation trees,  $c(n)$  is the average path length of a futile search, and  $n$  is the number of external nodes (Liu et al., 2008).

The Isolation Forest algorithm was applied to three types of data: (1) the raw action sequences (after vectorizing the actions with one-hot encoding), (2) the contextual embeddings obtained from BERT, and (3) the lower-dimensional contextual embeddings based on the first five principal components extracted from PCA. The first five principal components retained 85%–92% of the total variance in the embeddings. We used the *ensemble.IsolationForest* function from the *sklearn* library (Pedregosa et al., 2011) in Python to implement the Isolation Forest algorithm. In the function, contamination



rate is a hyperparameter that allows the user to manually specify a particular rate of anomalous cases. In this study, contamination rate was set to *auto* to determine the optimal rate for anomaly detection. This option allowed us to search for the optimal rate of anomalous cases for each item rather than extracting a fixed proportion of anomalous cases across all items. This is one of the advantages of employing Isolation Forest for anomaly detection because the user may not have conceptual or theoretical evidence regarding the rate of anomalous responses in the dataset (see Gorgun & Bulut, 2022).

The three anomaly detection methods were applied to the action sequences from each item in the PSTRE domain separately. Our codes for preprocessing the action sequences via BERT and detecting anomalous cases via Isolation Forest are available at <https://osf.io/cdm9t/>. For each method, we first explored the characteristics of anomalous and regular test-takers based on test-related outcomes, such as average sequence length, and response times. Next, we examined how test-takers' PSTRE proficiency levels differed based on their anomaly status (i.e., regular or anomalous test-taker). To account for error at the individual test-taker level, PIAAC reports 10 plausible values of proficiency (i.e., multiple imputations drawn from a posteriori distribution) rather than a single estimate of proficiency (for details about how plausible values are generated, refer to OECD, 2019). As the plausible values indicate similar patterns, we only used the first plausible value (PV1) to examine how anomaly flags derived from our analysis and PSTRE proficiency levels in PIAAC 2012 were related. Finally, we reviewed the relationship between the

anomaly status and several variables from the background questionnaire in PIAAC 2012.

## Results

### The Number of Anomalous Cases

Table 2 shows the number of anomalous test-takers identified by the three anomaly detection methods. The Isolation Forest algorithm based on the raw action sequences (iForest) and the lower-dimensional contextual embeddings (BERT + PCA + iForest) identified roughly 10% of the test-takers as anomalous for each PSTRE item. However, the number of anomalous cases identified by the same algorithm based on the contextual embeddings (BERT + iForest) varied significantly across the items (ranging from 8.74% to 21.28%). This method identified more than 18% of the test-takers as anomalous in the items with a high average sequence length (e.g., U04A, U02, and U16). The higher variability observed in the results based on contextual embeddings (BERT + iForest) may be attributed to the inherent complexity introduced by the high dimensionality of the data (768 dimensions), potentially leading to increased false positives or a greater sensitivity to action-specific characteristics. Notably, when employing lower-dimensional contextual embeddings (BERT + PCA + iForest), the algorithm exhibited a more consistent identification rate across items, indicating that the additional dimensionality reduction step might have contributed to stabilizing its performance.

**Table 2.** The percentages of anomalous cases by anomaly detection methods

Booklet	Item ID	Task name	<i>n</i>	% of anomalous cases		
				iForest	BERT + iForest	BERT + PCA + iForest
1	U01A	Party invitations	1,337	10.02	9.72	10.02
1	U01B	Party invitations	1,334	10.04	10.04	10.04
1	U03A	CD Tally	1,333	10.05	10.13	10.05
1	U06A	Sprained ankle	1,330	10.00	12.71	9.32
1	U06B	Sprained ankle	1,329	10.01	18.96	10.01
1	U21	Tickets	1,329	10.01	14.15	10.01
1	U04A	Class attendance	1,329	10.01	20.84	10.01
2	U19A	Club membership	1,340	10.00	8.88	10.00
2	U19B	Club membership	1,340	10.00	10.75	10.00
2	U07	Book order	1,340	10.00	12.69	9.85
2	U02	Meeting room	1,340	10.00	18.58	10.00
2	U16	Reply all	1,339	10.01	21.28	10.01
2	U11B	Locate email	1,338	10.01	10.76	10.01
2	U23	Lamp return	1,338	10.01	8.74	8.89

Note. *N* = The number of valid test-takers for each PSTRE item.

For each method, we also created an UpSet plot with the UpSetR package (Conway et al., 2017) in R to show the intersections (i.e., overlap) among the PSTRE items in terms of the number of anomalous test-takers (see Figures 4, 5, and 6). In the UpSet plot, each bar represents the number of anomalous cases (see the values above the bars), each row corresponds to a particular PSTRE item (sorted by the number of anomalous cases), and the filled-in cells indicate unique observations (detached dots) and intersections (dots connected vertically with others). For example, 59 test-takers were flagged as anomalous only in item U01A, whereas eight test-takers were flagged as anomalous in both items U01A and U21 (see Figure 4 for Isolation Forest based on raw action sequences). Figures 4, 5, and 6 show similar patterns in terms of the number of unique and intersecting anomalous cases across the

PSTRE items. The detached dots with high bars on the left part of the plot indicate that most test-takers were only identified as anomalous in one of the PSTRE items, whereas a smaller number of test-takers were flagged as an anomaly in either two or three items. None of the test-takers had anomalous action sequences in more than three PSTRE items.

### Anomaly Status and Test-Related Outcomes

Tables 3, 4, and 5 show a descriptive summary of test-related outcomes by anomaly status for each method. Several patterns deserve to be scrutinized to better understand these descriptive results. First, the anomalous test-takers generally had longer action sequences and

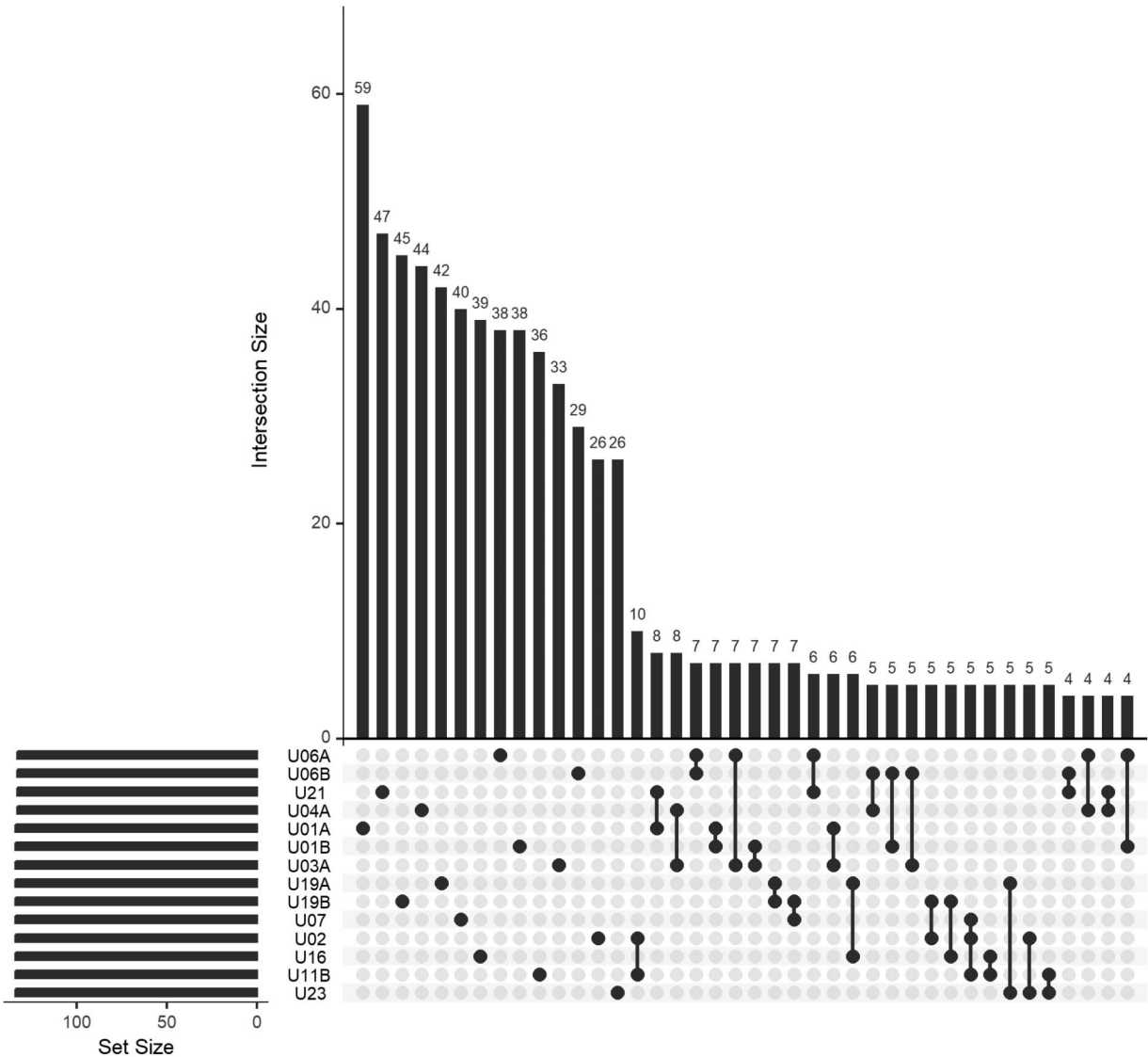
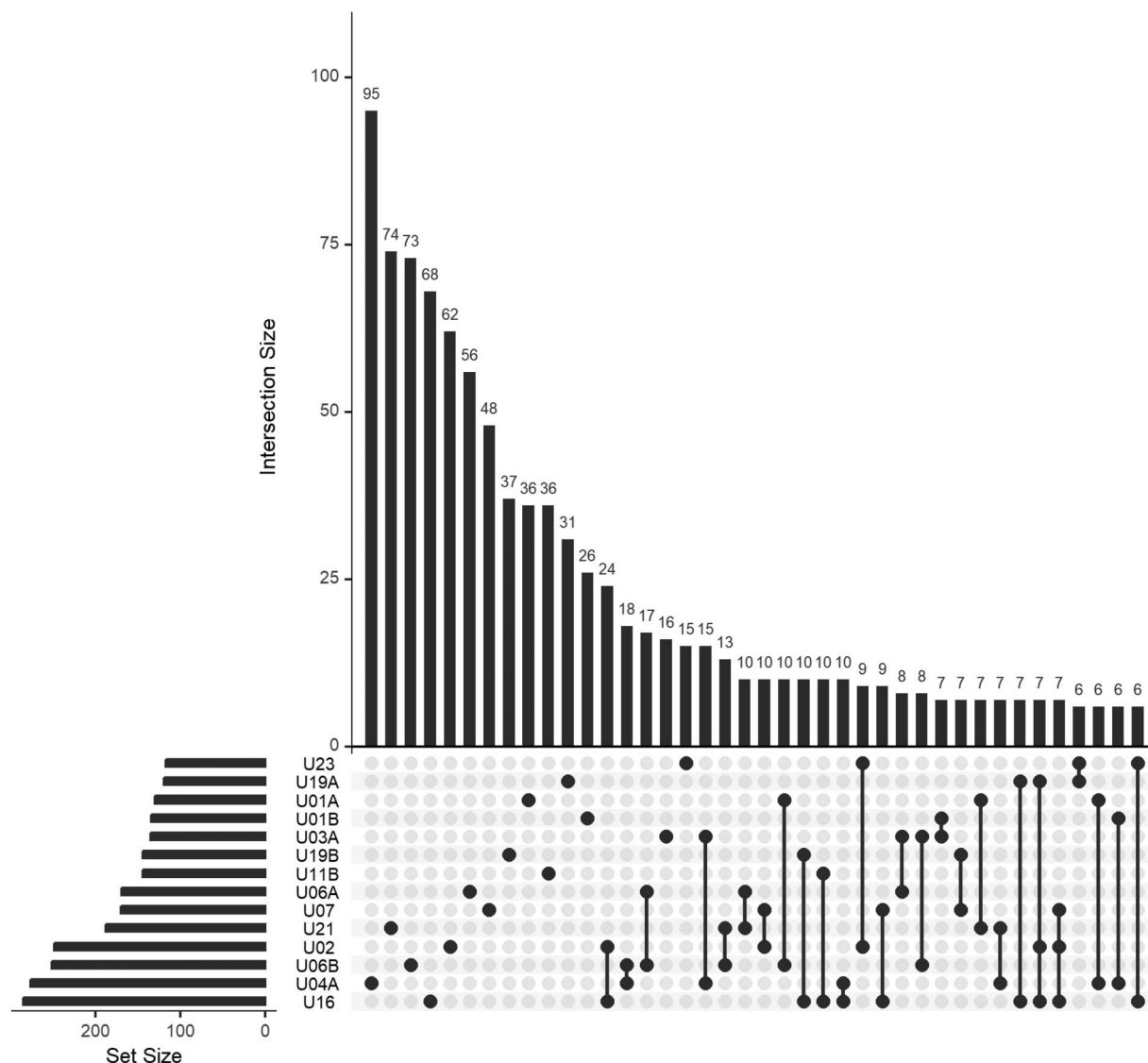


Figure 4. Anomalous cases across the 14 PSTRE items in Isolation Forest.

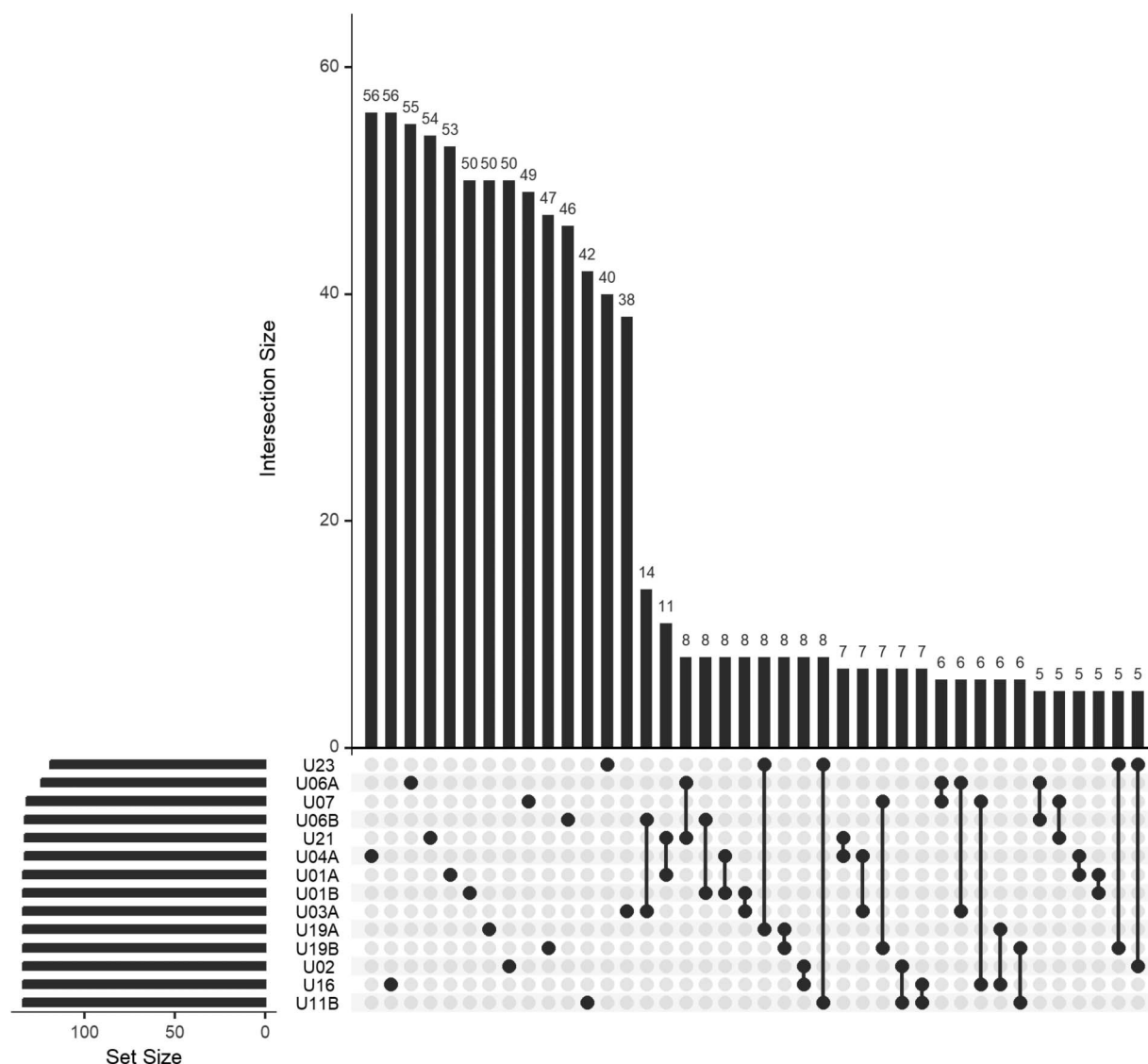


**Figure 5.** Anomalous cases across the 14 PSTRE items in BERT + Isolation Forest.

higher response times than the regular test-takers across all PSTRE items, except for item U04 in the lower-dimensional contextual embeddings (iForest + PCA + BERT). For example, for U03A in the first booklet and U11B in the second booklet, the average sequence lengths for the anomalous test-takers were at least three times longer than those for the regular test-takers. Second, the average response times for the anomalous test-takers were substantially higher than those for the regular test-takers, except for item U03A in the raw action sequences (iForest). The differences between the anomalous and regular test-takers in terms of their average sequence length and average response time seemed highly correlated ( $r > 0.73$ ) for all three methods. This was an anticipated finding because the test-takers with longer action sequences were

likely to spend more time on the PSTRE items than the test-takers with shorter action sequences.

The third pattern was about the average response accuracy in the PSTRE items. The results showed that the percentages of incorrect responses for the anomalous test-takers were generally lower than those for the regular test-takers (two to six items in the first booklet and six items in the second booklet). In addition, the percentages of incorrect responses for the regular test-takers indicated a moderate, positive correlation with the average sequence length ( $r > 0.45$ ) and the average response time ( $r > 0.47$ ). This finding suggests that the regular test-takers having more actions and spending more time on the items had a higher probability of answering the items incorrectly. However, these relationships were quite different for the



**Figure 6.** Anomalous cases across the 14 PSTRE items in BERT + PCA + Isolation Forest.

anomalous test-taker group. There was a moderate, negative correlation (around  $r = -0.3$ ) between the percentages of incorrect responses and the average sequence length, suggesting that the anomalous test-takers with longer action sequences were less likely to answer the items incorrectly. Also, unlike the regular test-taker group, the correlation between the percentages of incorrect responses and the average response time was relatively weak (around  $r = 0.15$ ) for the anomalous test-takers.

### Anomaly Status and Proficiency Level

We also examined the relationship between anomaly status and PSTRE proficiency levels (PV1). Tables 6, 7, and 8 show a descriptive summary of PV1 by anomaly

status and the results of independent samples  $t$ -tests comparing proficiency levels of anomalous and regular test-takers. For all three methods, the average PV1 values for the anomalous test-takers seemed to vary across the items due to different samples being flagged as anomalous. In contrast, the average PV1 values were mostly similar for the regular test-takers. The anomalous test-takers outperformed the regular test-takers in 13 PSTRE items in the raw action sequences and 10 items in the contextual embeddings. The difference between the two groups was statistically significant in all PSTRE items, except for three items (U01A, U06B, and U07) in the contextual embeddings and two items (U19A and U19B) in the lower-dimensional contextual embeddings. The larger differences in PV1 were mostly observed in the PSTRE items with longer action sequences (e.g., U04A,

**Table 3.** A descriptive summary of test-related outcomes by anomaly status based on Isolation Forest

Item ID	M (SD) sequence length		M (SD) response time		Incorrect response %	
	Anomalous	Regular	Anomalous	Regular	Anomalous	Regular
U01A	89.16 (32.59)	32.63 (12.62)	274.23 (181.00)	103.92 (80.21)	25.37	29.51
U01B	162.64 (55.27)	57.53 (33.22)	315.46 (183.73)	134.53 (82.66)	43.28	58.17
U03A	271.44 (220.39)	28.24 (16.91)	110.17 (51.69)	112.15 (82.12)	19.40	68.14
U06A	36.91 (10.96)	17.17 (3.85)	204.73 (99.94)	132.32 (91.82)	80.00	72.60
U06B	73.59 (16.07)	32.90 (16.32)	214.89 (100.80)	96.15 (73.73)	36.84	51.76
U21	69.58 (16.66)	37.68 (12.55)	285.27 (119.84)	175.18 (92.10)	57.90	61.79
U04A	447.69 (97.55)	86.33 (87.39)	609.93 (251.90)	265.72 (274.65)	62.41	86.87
U19A	103.78 (28.33)	41.84 (18.75)	182.56 (109.37)	118.42 (77.95)	18.66	37.14
U19B	145.14 (45.02)	43.58 (26.44)	388.24 (288.96)	173.09 (11.88)	20.90	38.23
U07	106.80 (84.96)	39.74 (21.86)	230.29 (76.66)	99.37 (65.54)	20.15	57.21
U02	343.57 (188.61)	75.12 (70.00)	459.05 (165.46)	188.63 (195.01)	23.39	74.63
U16	338.72 (117.65)	79.18 (60.19)	319.74 (137.79)	115.09 (84.56)	16.42	50.46
U11B	132.03 (52.31)	32.87 (23.76)	345.67 (1,028.63)	77.33 (59.50)	82.09	60.38
U23	147.73 (88.44)	40.54 (24.67)	218.22 (120.64)	86.17 (91.95)	24.63	61.71

**Table 4.** A descriptive summary of test-related outcomes by anomaly status based on BERT and Isolation Forest

Item ID	M (SD) sequence length		M (SD) response time		Incorrect response %	
	Anomalous	Regular	Anomalous	Regular	Anomalous	Regular
U01A	88.48 (34.49)	32.89 (12.95)	268.46 (188.27)	105.24 (81.14)	26.15	29.41
U01B	163.24 (54.79)	57.46 (33.10)	317.96 (177.91)	134.25 (83.45)	46.27	57.83
U03A	106.74 (54.71)	28.55 (17.44)	267.69 (221.60)	112.44 (82.22)	24.44	67.61
U06A	30.42 (15.07)	17.51 (3.89)	171.19 (111.13)	134.99 (91.75)	86.98	71.32
U06B	50.24 (29.09)	33.87 (16.23)	146.29 (116.40)	99.07 (72.58)	56.75	48.75
U21	44.94 (30.89)	40.20 (12.02)	195.00 (139.85)	184.74 (92.76)	83.51	57.76
U04A	171.14 (125.37)	109.68 (140.84)	548.06 (327.20)	235.16 (242.44)	88.05	87.74
U19A	104.39 (31.75)	42.54 (19.40)	189.90 (115.71)	118.49 (77.23)	21.85	35.38
U19B	143.10 (44.92)	42.97 (25.53)	424.14 (214.55)	166.97 (111.25)	20.14	38.46
U07	82.35 (84.96)	41.23 (22.79)	176.13 (93.72)	103.21 (70.13)	37.06	55.90
U02	206.62 (162.88)	78.08 (92.86)	393.45 (175.33)	175.09 (193.93)	41.37	75.89
U16	190.51 (92.03)	82.88 (93.89)	229.87 (118.44)	110.19 (92.46)	25.26	52.94
U11B	130.34 (51.03)	32.24 (22.79)	341.21 (992.09)	75.62 (56.84)	82.64	60.13
U23	152.30 (94.43)	41.59 (25.47)	218.29 (115.32)	88.11 (94.42)	31.62	60.52

U02, U16, and U23). Figure 7 shows the boxplots of PSTRE proficiency levels by anomaly status for each method. The median PSTRE proficiency levels for anomalous test-takers were generally higher than those for regular test-takers across all items when the raw action sequences were used (iForest), whereas, with the contextual embeddings, there was a greater variation among the items regarding the median PSTRE proficiency levels for anomalous and regular test-takers. Regardless of the anomaly detection method, the anomalous test-takers appeared to have a narrower ability distribution than the regular test-takers.

Anomaly Status and Background Variables

To further explore the profiles of anomalous and regular test-takers in PIAAC 2012, we evaluated the relationship between anomaly status (1: anomalous test-taker; 0: regular test-taker) and several background variables. Specifically, we computed point-biserial correlations between anomaly status and the following variables: (1) ICT skill use at home (ICTHome), (2) ICT skill use at work (ICTWork), (3) PSTRE proficiency level (PV1), (4) log of the number of actions in solving the item, (5) log of time to the first action in solving the item, and (6) log of total response time in solving the

**Table 5.** A descriptive summary of test-related outcomes by anomaly status based on BERT, PCA, and Isolation Forest

Item ID	<i>M (SD)</i> sequence length		<i>M (SD)</i> response time		Incorrect response %	
	Anomalous	Regular	Anomalous	Regular	Anomalous	Regular
U01A	75.99 (45.52)	34.09 (12.86)	235.31 (198.46)	103.47 (84.09)	35.07	28.43
U01B	145.19 (74.22)	59.38 (34.69)	293.99 (192.41)	136.93 (85.33)	59.70	56.33
U03A	110.05 (51.77)	28.25 (16.95)	270.73 (221.35)	112.23 (81.98)	20.90	67.97
U06A	32.06 (16.47)	17.82 (4.46)	172.81 (120.71)	136.18 (91.52)	88.71	71.72
U06B	65.66 (25.07)	33.79 (17.02)	200.54 (117.83)	97.94 (73.34)	46.62	50.67
U21	46.93 (34.29)	40.20 (12.48)	197.03 (148.46)	184.99 (93.99)	84.96	58.78
U04A	96.55 (37.13)	125.38 (146.74)	410.75 (292.82)	287.89 (288.67)	83.33	84.53
U19A	92.93 (40.80)	43.06 (19.78)	180.94 (113.75)	118.60 (77.40)	31.34	34.49
U19B	134.22 (61.43)	44.79 (27.17)	393.84 (228.12)	172.47 (118.92)	26.87	37.56
U07	75.32 (98.92)	43.29 (23.63)	159.15 (107.83)	107.36 (71.56)	59.09	52.90
U02	166.58 (152.86)	94.78 (113.78)	330.04 (163.28)	202.96 (209.31)	45.52	72.22
U16	125.34 (80.69)	102.91 (105.49)	169.76 (102.57)	131.82 (110.18)	35.82	48.30
U11B	128.89 (55.77)	33.22 (24.32)	344.88 (1,029.02)	77.42 (59.15)	79.10	60.71
U23	142.38 (100.81)	42.38 (26.43)	204.42 (120.90)	89.24 (95.31)	40.34	59.72

item. Tables 9, 10, and 11 show the correlations between anomaly status for each PSTRE item and the background variables. Two ICT-related variables (ICTHome and ICT-Work) generally indicated weak correlations with anomaly status for all three methods. The number of actions and response time indicated stronger associations with anomaly status. These correlations were negative for the raw action sequences (iForest) and the lower-dimensional contextual embeddings (BERT + PCA + iForest), suggesting that the test-takers with fewer actions and smaller response times

were more likely to be flagged as an anomaly. The correlations between anomaly status and time to the first action were also mostly negative but smaller in magnitude. However, these relationships were opposite for the contextual embeddings (BERT + iForest), suggesting that the test-takers with more actions, slower first reaction to the item, and slower response time were more likely to be flagged as anomalous. Overall, these findings suggest that the three methods are likely to flag different test-takers as anomalous depending on the representation (raw vs.

**Table 6.** A descriptive summary of PSTRE proficiency levels (PV1) by anomaly status based on Isolation Forest

Item ID	Anomalous test-takers				Regular test-takers				Mean difference	t-statistic
	<i>M</i>	<i>SD</i>	Min	Max	<i>M</i>	<i>SD</i>	Min	Max		
U01A	271.7	39.2	164.3	397.6	279.9	42.3	131.7	402.8	−8.2	−2.3*
U01B	296.2	31.5	191.2	354.3	277.2	42.6	131.7	402.8	18.9	6.3*
U03A	303.9	41.7	131.7	402.8	276.4	41.2	158.2	387.2	27.5	7.3*
U06A	288.3	34.6	206.5	387.2	278.1	42.7	131.7	402.8	10.2	3.1*
U06B	300.2	36.8	220.1	402.8	276.8	42.0	131.7	397.6	23.5	6.9*
U21	290.6	32.3	213.5	365.7	277.9	42.8	131.7	402.8	12.7	4.1*
U04A	313.3	26.5	257.4	382.5	275.3	41.8	131.7	402.8	38.0	14.7*
U19A	287.9	39.5	159.3	404.7	273.5	41.1	153.2	400.8	14.5	4.0*
U19B	285.0	32.2	192.7	379.0	273.8	41.9	153.2	404.7	11.3	3.7*
U07	295.3	31.3	210.9	384.7	272.6	41.5	153.2	404.7	22.7	7.7*
U02	308.3	26.5	228.3	384.7	271.3	40.9	153.2	404.7	36.0	14.0*
U16	294.8	29.6	210.9	367.4	272.7	41.7	153.2	404.7	22.1	7.8*
U11B	295.5	31.6	192.7	363.4	272.6	41.5	153.2	404.7	22.9	7.7*
U23	301.7	27.1	217.6	363.4	271.9	41.4	153.2	404.7	29.7	11.3*

Note. Mean difference = (PV1 for anomalous test-takers – PV1 for regular test-takers).

\* $p < .05$ . \*\* $p < .001$ .

**Table 7.** A descriptive summary of PSTRE proficiency levels (PV1) by anomaly status based on BERT and Isolation Forest

Item ID	Anomalous test-takers				Regular test-takers				Mean difference	t-statistic
	M	SD	Min	Max	M	SD	Min	Max		
U01A	273.3	38.1	164.3	346.4	279.7	42.4	131.7	402.8	−6.4	−1.8
U01B	296.6	31.9	191.2	379.1	277.2	42.6	131.7	402.8	19.4	6.4**
U03A	301.3	41.8	131.7	397.6	276.6	41.3	158.2	402.8	24.7	6.5**
U06A	268.8	41.6	158.2	381.3	280.6	41.9	131.7	402.8	−11.8	−3.5**
U06B	274.7	46.4	131.7	402.8	280.2	40.9	158.2	397.6	−5.5	−1.7
U21	263.2	38.2	159.3	246.6	281.7	42.1	131.7	402.8	−18.5	−6.1**
U04A	301.7	34.1	203.1	387.2	273.2	42.2	131.7	402.8	28.6	11.8**
U19A	282.3	39.8	159.3	404.7	274.2	41.2	153.2	400.8	8.1	2.1*
U19B	281.7	30.6	192.7	353.1	274.1	42.2	153.2	404.7	7.6	2.7*
U07	280.1	42.3	162.5	384.7	274.1	41.2	153.2	404.7	5.9	1.7
U02	299.2	31.8	183.8	404.7	269.4	41.1	153.2	400.8	29.8	12.6**
U16	293.3	31.4	192.7	369.7	269.9	42.1	153.2	404.7	23.4	10.3**
U11B	293.6	32.5	192.7	363.4	272.5	41.6	153.2	404.7	21.1	7.1**
U23	299.9	29.6	205.3	379.1	272.5	41.4	153.2	404.7	27.4	9.2**

Note. Mean difference = (PV1 for anomalous test-takers − PV1 for regular test-takers).  
\* $p < .05$ . \*\* $p < .001$ .

**Table 8.** A descriptive summary of PSTRE proficiency levels (PV1) by anomaly status based on BERT, PCA, and Isolation Forest

Item ID	Anomalous test-takers				Regular test-takers				Mean difference	t-statistic
	M	SD	Min	Max	M	SD	Min	Max		
U01A	266.6	37.4	164.3	397.6	280.4	42.3	131.7	402.8	−13.8	−4.0*
U01B	289.6	33.2	191.2	354.3	278.0	42.7	131.7	402.8	11.6	3.7*
U03A	304.0	40.8	131.7	397.6	276.4	41.3	158.2	402.8	27.6	7.4*
U06A	272.2	37.2	195.1	381.3	279.8	42.5	131.7	402.8	−7.6	−2.1*
U06B	290.6	41.5	191.6	402.8	277.8	41.9	131.7	397.6	12.7	3.4*
U21	261.4	38.7	159.3	346.6	281.1	42.0	131.7	402.8	−19.7	−5.5*
U04A	291.2	33.1	220.1	378.5	277.8	42.7	131.7	402.8	13.5	4.3*
U19A	276.9	40.4	159.3	404.7	274.7	41.3	153.2	400.8	2.3	0.6
U19B	278.2	32.2	153.2	326.8	274.5	42.0	158.2	404.7	3.7	1.2
U07	267.2	42.3	159.3	365.4	275.7	41.0	153.2	404.7	−8.5	−2.2*
U02	297.5	32.2	183.8	404.7	272.4	41.3	153.2	400.8	25.1	8.3*
U16	286.9	35.3	192.7	384.7	273.6	41.6	153.2	404.7	13.4	4.1*
U11B	295.1	32.0	211.3	363.4	272.6	41.5	153.2	404.7	22.5	7.5*
U23	295.7	31.8	183.8	353.1	272.9	41.5	153.2	404.7	22.9	7.3*

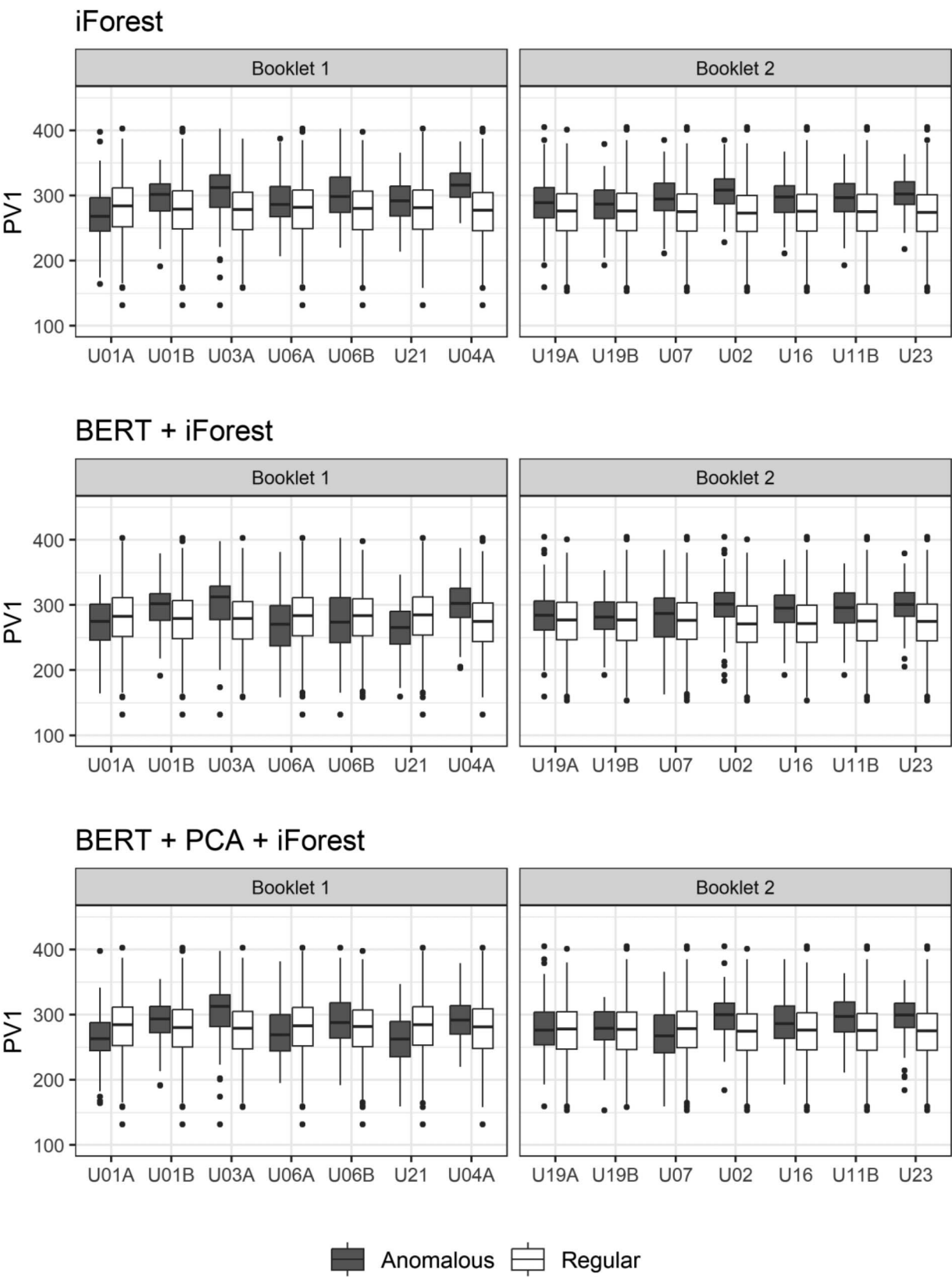
Note. Mean difference = (PV1 for anomalous test-takers − PV1 for regular test-takers).  
\* $p < .05$ . \*\* $p < .001$ .

embeddings) and dimensionality (high vs. low) of sequential process data.

## Discussion

This study evaluated the utility of unsupervised methods in detecting anomalous cases based on sequential process

data obtained from PIAAC 2012. Three forms of unsupervised anomaly detection with the Isolation Forest algorithm were demonstrated using different representations of sequential process data. In the first approach, the input data consisted of test-takers' raw action sequences represented by one-hot-encoding. The second approach involved extracting a 768-dimensional feature vector (i.e., contextual embeddings) using the BERT base model. The third approach employed a dimension



**Figure 7.** The distribution of PSTRE proficiency level (PV1) by anomaly status.



**Table 9.** Correlates of anomaly status for PSTRE items based on Isolation Forest

Item ID	ICT use at home	ICT use at work	PV1	Number of actions	Time to first action	Response time
U01A	-.080*	-.056*	.059*	-.665*	-.083*	-.474*
U01B	.049	.026	-.136*	-.574*	-.054*	-.487*
U03A	.033	.041	-.200*	-.675*	-.039	-.417*
U06A	.016	.051	-.073*	-.705*	.010	-.229*
U06B	.025	.040	-.168*	-.575	-.095*	-.421*
U21	.065*	.009	-.091*	-.518*	-.047	-.328*
U04A	.074*	.067*	-.271*	-.728*	-.060*	-.355*
U19A	.027	.028	-.106*	-.604*	.007	-.230*
U19B	.018	.012	-.082*	-.585*	-.098*	-.432*
U07	.077*	.023	-.165*	-.633*	-.078*	-.508*
U02	.077*	.038	-.262*	-.607*	.010	-.389*
U16	.069*	-.050	-.161*	-.504*	-.094*	-.559*
U11B	.077*	.054*	-.167*	-.642*	-.138*	-.238*
U23	.077*	.018	-.217*	-.645*	-.045	-.385*

Note. \* $p < .05$ .

**Table 10.** Correlates of anomaly status for PSTRE items based on BERT + Isolation Forest

Item ID	ICT use at home	ICT use at work	PV1	Number of actions	Time to first action	Response time
U01A	.076*	.054*	-.045	.311*	.074*	.423*
U01B	-.049	-.010	.139*	.272*	.101*	.351*
U03A	-.010	-.043	.177*	.431*	.088*	.352*
U06A	.046	-.014	-.094*	.149*	.067	.099*
U06B	.053*	.007	-.051	.041	.054	.073*
U21	.001	.041	-.154*	-.036	.081	-.003
U04A	-.061*	-.079*	.276*	.369*	.265*	.402*
U19A	-.033	-.038	.056*	.269*	.074	.212*
U19B	-.009	.015	.057*	.386*	.148*	.352*
U07	-.036	.003	.048	.233*	.121	.221*
U02	-.055*	-.096*	.282*	.384*	.194	.415*
U16	-.080*	-.076*	.233*	.371*	.184*	.397*
U11B	-.083*	-.036	.158*	.363*	.166*	.440*
U23	-.085*	-.035	.188*	.309*	.095	.341*

Note. \* $p < .05$ .

reduction technique (PCA) after extracting the 768-dimensional feature vector using BERT and yielded a lower-dimensional form of the contextual embeddings. To compare the results of these approaches, we analyzed the action sequences extracted from the PSTRE tasks included in PIAAC 2012. Our goal was to explore the characteristics of anomalous and regular test-takers identified by each method, investigate the relationship between anomaly status and test-takers' background characteristics and test-taking behaviors, and investigate how these characteristics differed based on the data processing step we adopted (i.e., raw action sequences, high-dimensional contextual embeddings, or lower-

dimensional contextual embeddings). We found consistent results with previous studies (Guo et al., 2021; Li et al., 2022) that a large language model, such as BERT, can help extract features from action sequence data, which can be used for further investigations, such as anomaly detection.

As previously mentioned, the term *anomaly* is often used to describe a negative situation based on an abnormal or peculiar case deviating from the accepted norms. From this definition, anomaly detection with action sequences from the PSTRE items might be expected to highlight test-takers with undesirable test-taking behaviors, such as careless or disengaged responding. However, the results of our study

**Table 11.** Correlates of anomaly status for PSTRE items based on BERT + PCA + Isolation Forest

Item ID	ICT use at home	ICT use at work	PV1	Number of actions	Time to first action	Response time
U01A	-.112*	-.046	.099*	-.510*	-.065*	-.351*
U01B	.058*	.021	-.083*	-.487*	-.060*	-.423*
U03A	.041	.046	-.198*	-.674*	-.04	-.415*
U06A	-.008	.034	.053	-.475*	.016	-.112*
U06B	.016	.030	-.091*	-.463*	-.087*	-.364*
U21	-.041	-.022	.140*	-.050	-.027	-.036
U04A	.025	.051	-.096*	.021	-.096*	-.127*
U19A	.018	.012	-.017	-.520*	-.021	-.223*
U19B	-.015	-.024	-.027	-.590*	-.126*	-.445*
U07	-.008	-.012	.062*	-.269*	-.044	-.200*
U02	.035	.049	-.183*	-.187*	.000	-.183*
U16	.035	.054*	-.098*	-.154*	-.038	-.103*
U11B	.060*	.023	-.164*	-.677	-.132*	-.237*
U23	.068*	.039	-.158*	-.546*	-.036*	-.317*

Note. \* $p < .05$ .

showed that anomalous test-takers differed from regular ones by taking more actions executed in a longer period of time when solving the PSTRE items. The number of anomalous cases was particularly higher in the PSTRE items that required a large number of actions on average (e.g., 20.84% in U04A – Class Attendance and 21.29% in U16 – Reply All). This finding is congruent with the previous research suggesting that disengaged responses could also be instantiated as idle responding (Gorgun & Bulut, 2023) where a test-taker spends unexpectedly longer response times, possibly leading to not-reached items (e.g., Bulut & Gorgun, 2023a; Gorgun & Bulut, 2021). Our study indicated that not only longer response times but also longer action sequences may be associated with anomalous responses. Additionally, the anomalous test-takers were generally less likely to answer the items incorrectly than the regular test-takers, suggesting that anomalous test-takers can be characterized by conducting many actions and using a long time to answer the item correctly. These results corroborate the findings of Ulitzsch, He, and Pohl (2022) who also analyzed the data from PIAAC 2012 and found that test-takers with incorrect answers indicated a longer exploration behavior but conducted fewer key actions in U01a – Party Invitations.

Consistent with the previous studies (e.g., He et al., 2019; Ulitzsch, He, & Pohl, 2022), our results also showed that behavioral patterns in the action sequences underlying anomaly status were associated with different proficiency levels in the PSTRE test. Compared to regular test-takers whose average proficiency levels, the average proficiency levels of anomalous test-takers varied more substantially across the items. Furthermore, anomalous

test-takers outperformed regular test-takers in most of the PSTRE items, suggesting that the anomalous group might have followed a more effective strategy in solving the items. This finding also underlines the importance of investigating the profile of anomalous test-takers for better differentiating between test-takers who performed well in the PSTRE test.

To further describe the profile of anomalous test-takers, we examined the relationship between anomaly status and test-takers' background variables. Previous studies found that demographic variables such as ICT skills use at home or work were moderately associated with test-takers' action sequences and proficiency levels in the PSTRE items (e.g., He et al., 2019; Liao et al., 2019; Zhang et al., 2021). However, the findings of the current study did not support previous research. ICT-related variables used in the current study (i.e., ICT skill use at home or work) indicated weak correlations with anomaly status. Unlike ICT-related variables, response time, number of actions, and time to the first action indicated stronger correlations with anomaly status. When either raw or lower-dimensional contextual embeddings were used, test-takers who showed shorter exploration behavior prior to their first action, completed fewer actions, and spent less time in solving the items were more likely to be identified as anomalous. However, these relationships were entirely opposite when the original contextual dimensions extracted from BERT (with 768 dimensions) were used as the input data for the Isolation Forest algorithm. These findings suggest that the Isolation Forest algorithm was sensitive to the dimensionality of the input data as the length of the input vector appeared to

influence the determination of norm (i.e., regular) and deviant (i.e., anomalous) groups.

## Limitations and Future Research

Despite the novelty of combining a large language model with an anomaly detection algorithm to analyze sequential process data, some limitations are also worth discussing. First, this study only considered test-takers' action sequences in the PSTRE items. However, the log file for the PSTRE items also includes time stamps that may provide additional information on the actions (e.g., the time required for executing each action and the time elapsed between actions). Previous studies indicated that time stamps combined with action sequences can reveal unique insights about behavioral patterns in digital problem-solving tasks (Ulitzsch, He, & Pohl, 2022). Hence, combining the sequence embeddings with time stamps would be worthwhile before running an anomaly detection algorithm. Second, this study used the BERT model to transform raw action sequences into contextual embeddings. As described earlier, the BERT model processes a maximum input length of 512 tokens and truncates longer sequences beyond this limit. Future research is needed to explore the utility of action sequence embeddings based on other language models capable of processing longer sequences (e.g., Longformer; Beltagy et al., 2020). Finally, this exploratory study focused on anomaly detection based on action sequences in the PSTRE items. However, we could not discuss the results in relation to the tasks presented in the items due to not having access to the documentation regarding nonreleased PSTRE items in PIAAC 2012. Future studies can investigate the role of item content or task type in anomaly detection by using problem-solving items from other digital assessments, such as the National Assessment of Educational Progress (NAEP).

## References

- Beltagy, I., Peters, M. E., & Cohan, A. (2020). *Longformer: The long-document transformer*. arXiv preprint arXiv:2004.05150. <https://doi.org/10.48550/arXiv.2004.05150>
- Bulut, O., & Gorgun, G. (2023a, June). *Utilizing response time for scoring the TIMSS 2019 problem solving and inquiry tasks*. Paper presented at the 10th IEA International Research Conference (IEA IRC), Dublin, Ireland. <https://doi.org/10.31234/osf.io/zc98s>
- Bulut, O., & Gorgun, G. (2023b, November). *Unsupervised Anomaly Detection in Sequential Process Data: Insights from PIAAC Problem-Solving Tasks* [Supplementary Materials]. <https://doi.org/10.17605/OSF.IO/CDM9T>
- Catillo, M., Pecchia, A., & Villano, U. (2022). AutoLog: Anomaly detection by deep autoencoding of system logs. *Expert Systems with Applications*, 191, Article 116263. <https://doi.org/10.1016/j.eswa.2021.116263>
- Chen, S., & Liao, H. (2022). BERT-Log: Anomaly detection for system logs based on pre-trained language model. *Applied Artificial Intelligence*, 36(1), Article 2145642. <https://doi.org/10.1080/08839514.2022.2145642>
- Chen, Y., Zhang, J., Yang, Y., & Lee, Y. (2022). Latent space model for process data. *Journal of Educational Measurement*, 59(4), 517–535. <https://doi.org/10.1111/jedm.12337>
- Conway, J. R., Lex, A., & Gehlenborg, N. (2017). UpSetR: an R package for the visualization of intersecting sets and their properties. *Bioinformatics*, 33(18), 2938–2940. <https://doi.org/10.1093/bioinformatics/btx364>
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). *BERT: Pre-training of deep bidirectional transformers for language understanding*. arXiv preprint arXiv:1810.04805. <https://doi.org/10.48550/arXiv.1810.04805>
- Gao, Y., Cui, Y., Bulut, O., Zhai, X., & Chen, F. (2022). Examining adults' web navigation patterns in multi-layered hypertext environments. *Computers in Human Behavior*, 129, Article 107142. <https://doi.org/10.1016/j.chb.2021.107142>
- Goldhammer, F., Naumann, J., Roelke, H., Stelter, A., & Toth, K. (2017). Relating product data to process data from computer-based competency assessment. In D. Leutner, J. Fleischer, J. Grünkorn, & E. Klieme (Eds.), *Competence assessment in education: Research, models and instruments* (pp. 407–425). Springer. [https://doi.org/10.1007/978-3-319-50030-0\\_24](https://doi.org/10.1007/978-3-319-50030-0_24)
- Gómez-Alonso, C., & Valls, A. (2008). A similarity measure for sequences of categorical data based on the ordering of common elements. In V. Torra & Y. Narukawa (Eds.), *Modeling decisions for artificial intelligence* (pp. 134–145). Springer. [https://doi.org/10.1007/978-3-540-88269-5\\_13](https://doi.org/10.1007/978-3-540-88269-5_13)
- Gorgun, G., & Bulut, O. (2021). A polytomous scoring approach to handle not-reached items in low-stakes assessments. *Educational and Psychological Measurement*, 81(5), 847–871. <https://doi.org/10.1177/0013164421991211>
- Gorgun, G., & Bulut, O. (2022). Identifying aberrant responses in intelligent tutoring systems: An application of anomaly detection methods. *Psychological Test and Assessment Modeling*, 64(4), 359–384. <https://doi.org/10.1002/9781119439004.ch16>
- Gorgun, G., & Bulut, O. (2023). Incorporating test-taking engagement into the item selection algorithm in low-stakes computerized adaptive tests. *Large-scale Assessments in Education*, 11(1), Article 27. <https://doi.org/10.1186/s40536-023-00177-5>
- Guo, H., Yuan, S., & Wu, X. (2021, July). *Logbert: Log anomaly detection via bert*. The International Joint Conference on Neural Networks (IJCNN), Shenzhen, China (pp. 1–8). <https://doi.org/10.1109/IJCNN52387.2021.9534113>
- Hahnel, C., Ramalingam, D., Kroehne, U., & Goldhammer, F. (2022). Patterns of reading behaviour in digital hypertext environments. *Journal of Computer Assisted Learning*, 39(3), 737–750. <https://doi.org/10.1111/jcal.12709>
- He, Q., Borgonovi, F., & Paccagnella, M. (2021). Leveraging process data to assess adults' problem-solving skills: Using sequence mining to identify behavioral patterns across digital tasks. *Computers & Education*, 166, Article 104170. <https://doi.org/10.1016/j.compedu.2021.104170>
- He, Q., Liao, D., & Jiao, H. (2019). Clustering behavioral patterns using process data in PIAAC problem-solving items. In B. Veldkamp & C. Sluijter (Eds.), *Theoretical and practical advances in computer-based educational measurement* (pp. 189–212). Springer. [https://doi.org/10.1007/978-3-030-18480-3\\_10](https://doi.org/10.1007/978-3-030-18480-3_10)

- He, Q., Meadows, M., & Black, B. (2022). An introduction to statistical techniques used for detecting anomaly in test results. *Research Papers in Education*, 37(1), 115–133. <https://doi.org/10.1080/02671522.2020.1812108>
- He, Q., & Davier, M. von. (2015). Identifying feature sequences from process data in problem-solving items with n-grams. In L. A. van der Ark, D. M. Bolt, W.-C. Wang, J. A. Douglas, & S.-M. Chow (Eds.), *Quantitative psychology research: The 79th Annual Meeting of the Psychometric Society* (pp. 173–190). Springer. [https://doi.org/10.1007/978-3-319-19977-1\\_13](https://doi.org/10.1007/978-3-319-19977-1_13)
- Hu, Y., Wu, B., & Gu, X. (2017). Learning analysis of K-12 students' online problem solving: A three-stage assessment approach. *Interactive Learning Environments*, 25(2), 262–279. <https://doi.org/10.4324/9780429428500-10>
- Iglesias, F., & Zseby, T. (2015). Analysis of network traffic features for anomaly detection. *Machine Learning*, 101, 59–84. <https://doi.org/10.1007/s10994-014-5473-9>
- Johns, J., & Woolf, B. (2006, July). A dynamic mixture model to detect student motivation and proficiency. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference* (pp. 163–168), Boston, MA, USA.
- Kamalov, F., Sulieman, H., & Santandreu Calonge, D. (2021). Machine learning based approach to exam cheating detection. *PLoS ONE*, 16(8), Article e0254340. <https://doi.org/10.1371/journal.pone.0254340>
- Karlsen, E. (2023). *Exploration of NLP-based feature extraction techniques for security analysis and anomaly detection of service logs* [Doctoral dissertation, Dalhousie University]. <http://hdl.handle.net/10222/82552>
- Kim, D., Woo, A., & Dickison, P. (2016). Identifying and investigating aberrant responses using psychometrics-based and machine learning-based approaches. In G. J. Cizek & J. A. Wollack (Eds.), *Handbook of quantitative methods for detecting cheating on tests* (pp. 70–97). Routledge. <https://doi.org/10.4324/9781315743097-4>
- Landauer, M., Onder, S., Skopik, F., & Wurzenberger, M. (2023). Deep learning for anomaly detection in log data: A survey. *Machine Learning with Applications*, 12, Article 100470. <https://doi.org/10.1016/j.mlwa.2023.100470>
- Le, V. H., & Zhang, H. (2021, November). Log-based anomaly detection without log parsing. In *2021 36th IEEE/ACM International conference on Automated Software Engineering (ASE)* (pp. 492–504). IEEE. <https://doi.org/10.1109/ASE51524.2021.9678773>
- Li, X., Cheng, K., Huang, T., & Tan, S. (2022). Research on false alarm detection algorithm of nuclear power system based on BERT-SAE-iForest combined algorithm. *Annals of Nuclear Energy*, 170, Article 108985. <https://doi.org/10.1016/j.anucene.2022.108985>
- Liao, D., He, Q., & Jiao, H. (2019). Mapping background variables with sequential patterns in problem-solving environments: An investigation of us adults' employment status in PIAAC. *Frontiers in Psychology*, 10, Article 646. <https://doi.org/10.3389/fpsyg.2019.00646>
- Liao, M., Patton, J., Yan, R., & Jiao, H. (2021). Mining process data to detect aberrant test takers. *Measurement: Interdisciplinary Research and Perspectives*, 19(2), 93–105. <https://doi.org/10.1080/15366367.2020.1827203>
- Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining* (pp. 413–422). <https://doi.org/10.1109/ICDM.2008.17>
- Meena Siwach, D., & Mann, S. (2022). Anomaly detection for web log data analysis: A review. *Journal of Algebraic Statistics*, 13(1), 129–148. <https://doi.org/10.52783/jas.v13i1.68>
- Nagy, G., Ulitzsch, E., & Lindner, M. A. (2022). The role of rapid guessing and test-taking persistence in modelling test-taking engagement. *Journal of Computer Assisted Learning*, 39(3), 751–766. <https://doi.org/10.1111/jcal.12719>
- Omar, S., Ngadi, A., & Jebur, H. H. (2013). Machine learning techniques for anomaly detection: An overview. *International Journal of Computer Applications*, 79(2), 33–41. <https://doi.org/10.5120/13715-1478>
- Organization for Economic Cooperation and Development [OECD]. (2019). *Technical report of the survey of adult skills (PIAAC)* (3rd ed.). OECD Publishing. [https://www.oecd.org/skills/piaac/publications/PIAAC\\_Technical\\_Report\\_2019.pdf](https://www.oecd.org/skills/piaac/publications/PIAAC_Technical_Report_2019.pdf)
- Pan, Y., & Choe, E. M. (2021). *An autoencoder-based response time model and its application in anomaly detection*. <https://doi.org/10.31234/osf.io/mw2y7>
- 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., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830. <https://doi.org/10.5555/1953048.2078195>
- R Core Team. (2022). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing.
- Rezapour, M. (2019). Anomaly detection using unsupervised methods: Credit card fraud case study. *International Journal of Advanced Computer Science and Applications*, 10(11), Article 101101. <https://doi.org/10.14569/IJACSA.2019.0101101>
- Ryciak, P., Wasielewska, K., & Janicki, A. (2022). Anomaly detection in log files using selected natural language processing methods. *Applied Sciences*, 12(10), Article 5089. <https://doi.org/10.3390/app12105089>
- Shao, Y., Zhang, W., Liu, P., Huyue, R., Tang, R., Yin, Q., & Li, Q. (2022, April). Log anomaly detection method based on BERT model optimization. In IEEE. (Ed.), *The 7th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)* (pp. 161–166). IEEE. <https://doi.org/10.1109/ICCCBDA55098.2022.9778900>
- Shin, H. J., Jewsbury, P. A., & van Rijn, P. W. (2022). Generating group-level scores under response accuracy-time conditional dependence. *Large-Scale Assessments in Education*, 10, Article 4. <https://doi.org/10.1186/s40536-022-00122-y>
- Stadler, M., Fischer, F., & Greiff, S. (2019). Taking a closer look: An exploratory analysis of successful and unsuccessful strategy use in complex problems. *Frontiers in Psychology*, 10, Article 777. <https://doi.org/10.3389/fpsyg.2019.00777>
- Tang, X., Wang, Z., He, Q., Liu, J., & Ying, Z. (2020). Latent feature extraction for process data via multidimensional scaling. *Psychometrika*, 85(2), 378–397. <https://doi.org/10.1007/s11336-020-09708-3>
- Tang, X., Wang, Z., Liu, J., & Ying, Z. (2021). An exploratory analysis of the latent structure of process data via action sequence autoencoders. *British Journal of Mathematical & Statistical Psychology*, 74(1), 1–33. <https://doi.org/10.1111/bmsp.12203>
- Ulitzsch, E., He, Q., & Pohl, S. (2022). Using sequence mining techniques for understanding incorrect behavioral patterns on interactive tasks. *Journal of Educational and Behavioral Statistics*, 47(1), 3–35. <https://doi.org/10.3102/10769986211010467>
- Ulitzsch, E., Yildirim-Erbasli, S. N., Gorgun, G., & Bulut, O. (2022). An explanatory mixture IRT model for careless and insufficient effort responding in self-report measures. *British Journal of Mathematical and Statistical Psychology*, 75(3), 668–698. <https://doi.org/10.1111/bmsp.12272>
- Van der Linden, W. J., & Jeon, M. (2012). Modeling answer changes on test items. *Journal of Educational and Behavioral Statistics*, 37(1), 180–199. <https://doi.org/10.3102/1076998610396899>
- Viswanathan, S. A., & Vanlehn, K. (2017). *High accuracy detection of collaboration from log data and superficial speech features*. 12th

- International Conference on Computer Supported Collaborative Learning (Vol. 1, pp. 335–342). <https://repository.isls.org/handle/1/249>
- Wang, Z., & AnilKumar, A. (2023). *W2R: An ensemble anomaly detection model inspired by language models for web application firewalls security* [Master thesis, Halmstad University]. <https://urn.kb.se/resolve?urn=urn%3Anbn%3Ase%3Ahh%3Adiva-50921>
- Wittkopp, T., Acker, A., Nedelkoski, S., Bogatinovski, J., Scheinert, D., Fan, W., & Kao, O. (2021). *A2log: attentive augmented log anomaly detection*. arXiv preprint arXiv:2109.09537. <https://doi.org/10.48550/arXiv.2109.09537>
- Xiao, Y., He, Q., Veldkamp, B., & Liu, H. (2021). Exploring latent states of problem-solving competence using hidden Markov model on process data. *Journal of Computer Assisted Learning*, 37(5), 1232–1247. <https://doi.org/10.1111/jcal.12559>
- Xu, H., Fang, G., Chen, Y., Liu, J., & Ying, Z. (2018). Latent class analysis of recurrent events in problem-solving items. *Applied Psychological Measurement*, 42(6), 478–498. <https://doi.org/10.1177/0146621617748325>
- Zhang, S., Tang, X., He, Q., Liu, J., & Ying, Z. (2021). *External correlates of adult digital problem-solving behavior: Log data analysis of a large-scale assessment*. arXiv preprint arXiv:2103.15036. <https://doi.org/10.48550/arXiv.2103.15036>
- Zhu, M., Zhang, M., & Deane, P. (2019). Analysis of keystroke sequences in writing logs. *ETS Research Report Series*, 2019(1), 1–16. <https://doi.org/10.1002/ets2.12247>

## History

Received June 1, 2022

Revision received November 17, 2023

Accepted December 11, 2023

Published online April 24, 2024

## Open Data

The supplementary materials are available at <https://osf.io/cdm9t/> (Bulut & Gorgun, 2023b).

## ORCID


Okan Bulut

 <https://orcid.org/0000-0001-5853-1267>

Guher Gorgun

 <https://orcid.org/0000-0002-0861-9225>

Surina He

 <https://orcid.org/0000-0002-9859-9749>

## Okan Bulut

Centre for Research in Applied Measurement and Evaluation

Faculty of Education

University of Alberta

6-110 Education Centre North, 11210 87 Avenue NW

Edmonton, AB T6G 2G5

Canada

[bulut@ualberta.ca](mailto:bulut@ualberta.ca)