Psychological Test and Assessment Modeling, Volume 64, 2022 (4), 385-424

A Machine Learning Approach for Detecting Item Compromise and Preknowledge in Computerized Adaptive Testing

Yiqin Pan[1], *Sandip Sinharay* [2] *Oren Livne*[2] *& James A. Wollack*[3]

Abstract

Item compromise and preknowledge have become common concerns in educational testing. We propose a machine learning approach to simultaneously detect compromised items and examinees with item preknowledge in computerized adaptive testing. The suggested approach provides a confidence score that represents the confidence that the detection result truly corresponds to item preknowledge and draws on ideas in ensemble learning, conducting multiple detections independently on subsets of the data and then combining the results. Each detection first classifies a set of responses as aberrant using a self-training algorithm and support vector machine, and identifies suspicious examinees and items based on the classification result. The confidence score is adapted, using the autoencoder algorithm, from the confidence score that Pan and Wollack (2022) suggested for non-adaptive tests. Simulation studies demonstrate that the proposed approach performs well in item preknowledge detection and the confidence score can provide helpful information for practitioners.

Keywords: test security, item preknowledge, machine learning, computerized adaptive testing, support vector machine, autoencoder

[1] University of Florida

[2] Educational Testing Service

[3] University of Wisconsin-Madison *Correspondence concerning this article should be addressed to:* Yiqin Pan, Research and Evaluation Methodology, College of Education, University of Florida, 1215 Norman Hall, Gainesville, FL 32611, USA. ypan@coe.ufl.edu

Item preknowledge refers to the phenomenon in which several examinees have access to several live/operational items prior to taking a test (Foster, 2013). The examinees who may have benefited from item preknowledge are referred to as *examinees with preknowledge* (EWP) and the leaked items are referred to as *compromised items* (CI). Preknowledge tends to provide unfair advantage to EWP and hence threatens the fairness and validity of scores. To protect test integrity, several methods have been developed to detect item preknowledge (e.g., Cizek & Wollack, 2017). A common detection strategy used in previous methods is to flag the examinees or items that manifest an aberrant pattern in responses or response times. For instance, an examinee may be flagged if he/she has a score vector that is unlikely under a hypothesized item response theory model (e.g., Belov & Armstrong, 2011; Drasgow, Levine, & Williams, 1985; McLeod, Lewis, & Thissen, 2003; Sinharay, 2017), and an item may be flagged if an unexpectedly high percentage of responses to the item are correct (e.g., Choe, Zhang & Chang, 2018; Liu, Han & Li, 2019; Zhang, 2014; Zhang & Li, 2016). Most of these methods are primarily theory-driven; that is, they are based on various sets of assumptions about how the data should behave under item preknowledge. Although useful in certain contexts, such models often inadequately represent the complexities of realistic testing situations and display poor fit to empirical data. For instance, several approaches (e.g., Wang, Xu, Shang, & Kuncel, 2018) use a hierarchical model (van der Linden, 2007) to modeling item responses and response times and flag examinees/items with aberrant patterns, but the hierarchical model has been found to not fit real data adequately (e.g., Domingue, Kanopka, Stenhaug, Soland, Kuhfeld, Wise, & Piech, 2021; Sinharay & van Rijn, 2020).

Machine learning (ML) algorithms learn from data to make predictions or decisions about unknown events without explicit instructions (Alpaydin, 2004). These methods can be generally categorized as either supervised learning methods or unsupervised learning methods (Alpaydin, 2004). In supervised learning, the algorithm learns a mapping function from input variables (e.g., examinees' response vectors) to output variables (e.g., aberrant response, normal response) given a labeled set of input-output pairs—this step is often referred to as the "training" of the algorithm and the labeled data[4] used in training is referred to as training data. After the training, the algorithm is used to predict an output (e.g., aberrant or normal) for a new set of unlabeled data (Kotsiantis, Zaharakis & Pintelas, 2007).

Unsupervised machine learning evaluates the similarity between and among different variables for purposes of finding special or interesting patterns, including latent groups or clusters, embedded in the data (e.g., Figueiredo & Jain, 2002). In unsupervised learning algorithms in the context of detection of test fraud, only input variables (e.g., item responses, response times, other process data, etc.) are evaluated for purposes of uncovering underlying patterns among the data (Längkvist, Karlsson & Loutfi, 2014).

[4] Labeled data are a group of samples that have tags, such as responses with an 'aberrant' tag. Likewise, unlabeled data are data without tags.

ML algorithms are not reliant on specific theories. Unlike statistical procedures (like regression methods) that intend to make theory-based inferences from samples, machine learning algorithms specialize in recognizing generalizable and predictive patterns. In other words, statistical methods concentrate on explicitly verifying assumptions about the problem and refining the specified models, or providing quantitative statements about the confidence for the models; machine learning methods focus on forecasting unseen outcomes, while making minimum assumptions about the data-generating process. Thus, statistical models are chosen based on our domain knowledge in statistics while machine learning models are chosen because of their empirical capabilities. Several researchers have successfully applied ML techniques to detect item preknowledge. Thomas (2016) used the *support vector machine* (SVM) model to detect CI on a certification exam for which approximately 60% of the items were suspected of being compromised, which resulted in 75% detection accuracy. Man, Harring, and Sinharay (2019) applied a series of ML methods to detect EWP in two data sets from licensure examinations. Zopluoglu (2019) used examinees flagged for preknowledge to train an extreme gradient boosting algorithm to detect EWP in large-scale testing. Zhou and Jiao (2022) investigated the application of the stacking ensemble machine learning method to detect cheating behaviors, using the item response and response time of examinees. Pan and Wollack (2021) proposed an unsupervised approach based on deep clustering to detect CI in non-adaptive/linear testing, which was able to classify well provided the amount of preknowledge is not overwhelming and aberrance effect is at least moderate. Although past studies provide strong evidence of the promise of ML methods in preknowledge detection, existing approaches are only designed for linear tests, and have not yet been extended to computerized adaptive tests (CATs) that are typically more susceptible to item preknowledge (e.g., McLeod, Lewis, & Thissen, 2003). In an attempt to fill this void, we develop a ML approach to simultaneously detect EWP and CI for CATs in this paper.

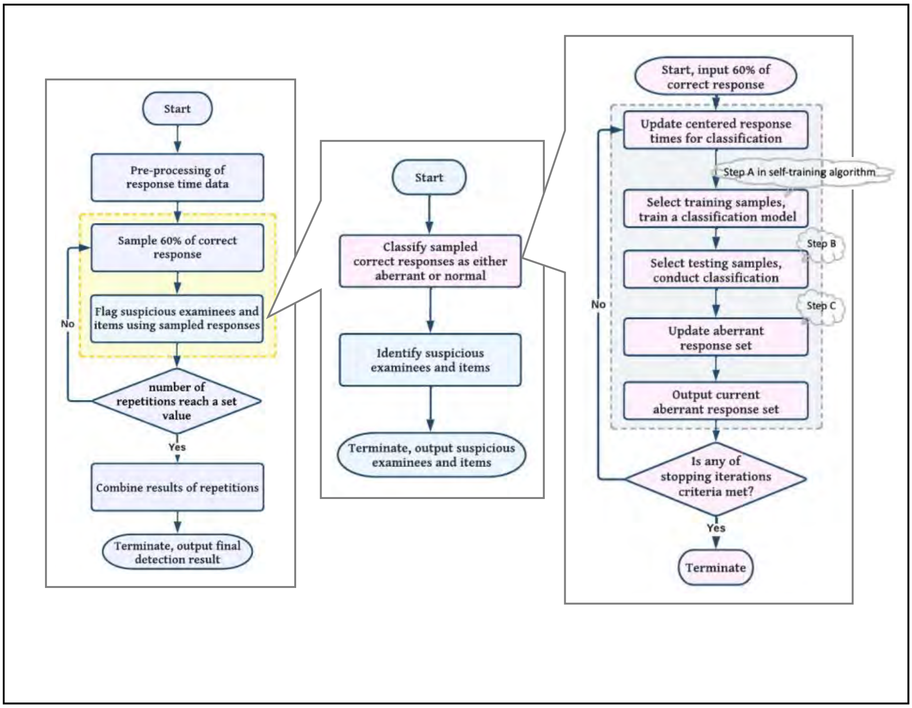
The ML approach flags suspicious examinees and items in two steps. In Step 1, a set of responses is classified as aberrant. A desired classification procedure involves first training a classifier using labeled data, and then applying the trained classifier to label the unlabeled data points. However, this paper considers cases where there are no labeled data for classifier training. Therefore, we draw on the ideas underlying the self-training algorithm (Zhu & Goldberg, 2009) to conduct the classification procedure. The self-training algorithm is a procedure that can label a large amount of unlabeled data with a small amount of training data and reduces the dependence on prior information about the compromise status. In implementing the classification procedure, we bring in domain knowledge to identify a small set of responses as aberrant/normal training data, and use SVM, a robust classification model (Suykens & Vandewalle, 1999), as the classifier. Generally, both SVMs and neural networks outperform other classification models when dealing with multi-dimensions and continuous features (Kotsiantis et al., 2007; Suykens & Vandewalle, 1999). Neural networks require a large amount of observed data in training, but the amount of our training data is small. Therefore, we use SVM as a classifier. In Step 2, the ML approach flags as suspicious the examinees and items appearing in the aberrant response set with a strong preknowledge signal.

However, if we detect preknowledge by repeating these two steps only once, the detection performance on different datasets may vary widely. Also, the extent of errors in the detection procedure may be reduced by repeating these two steps on different subsets of the dataset. Thus, we employ ensemble learning on top of this process (Sagi & Rokach, 2018), sampling multiple subsets of the data with replacement to flag the suspicious items and examinees separately and then combining the results to a final detection result.

After finishing the detection procedure, to assist practitioners in deciding whether to use the detection result, the ML approach also provides a confidence score adapted from Pan and Wollack (2022; PW22) to provide a measure of confidence that the detection result reflects actual item preknowledge. The rest of this paper includes descriptions of (a) the proposed detection algorithm; (b) the confidence score for CATs; (c) results from a simulation study to examine the performance of the ML approach; and (d) the implications and limitations of this study, and directions for future exploration.

Figure 1

Flowchart of the ML Algorithm


Start
Pre-processing of response time data
Sample 60% of correct response
Flag suspicious examinees and items using sampled responses
No
number of repetitions reach a set value
Yes
Combine results of repetitions
Terminate, output final detection result
Start
Classify sampled correct responses as either aberrant or normal
Identify suspicious examinees and items
Terminate, output suspicious examinees and items
Start, input 60% of correct response
Update centered response times for classification
Step A in self-training algorithm
Select training samples, train a classification model
Step B
Select testing samples, conduct classification
Step C
Update aberrant response set
Output current aberrant response set
No
Is any of stopping iterations criteria met?
Yes
Terminate

Note. The steps within the yellow frame constitute what we refer to as one *repetition*. The steps within the grey frame constitute what we refer to as one *iteration*.

Description of the Proposed Algorithm

The fundamental assumption underlying our methodology is that item preknowledge will manifest as EWP, producing responses to CI more quickly and more accurately than expected. Based on this assumption, the ML algorithm flags suspicious examinees and items in two steps 1) classification of a set of responses as aberrant; 2) identification of suspicious examinees and items using the classification result from Step 1. Steps 1-2 constitute a single *repetition*. Repetitions are performed until the number of repetitions reach a set value. To provide a convincing and reliable result, we incorporate ensemble learning into the ML approach. Figure 1 shows the workflow of our algorithm hierarchically. The left panel is the general flowchart of the ML approach. The middle panel provide a detailed flowchart for a single repetition, which is to flag suspicious examinee and item. The right panel includes the flowchart for step 1 in a single repletion, which classifies a set of responses as aberrant.

Overview of Step 1: Aberrant Response Classification

The first step of each repetition identifies aberrant responses, which are later used to flag suspicious examinees and items. According to our assumption that EWP's responses on CI are expected to have high accuracies, we argue that the aberrant responses, which can help to flag suspicious examinees and items, are likely to be correct. Thus, only correct responses are considered in the identification of aberrant responses.

To reduce the need for prior information about the compromise status of items and examinees, the classification procedure is developed within a self-training algorithm framework. The self-training algorithm is an iterative process, where, in each iteration, (1A) a classifier is trained based on the labeled data; (1B) the classifier is used to classify the unlabeled data; and (1C) the unlabeled points with the most confident classification result, which is a subset of unlabeled data, are added to the labeled data set. Iterations are performed until all data points are labeled. We will also introduce several modifications to the self-training algorithm so that it can be applied to the task of response classification.

In step 1A, since we do not assume any prior information about compromise status, there is no labeled data. Therefore, we modify the way to prepare training data. In the first iteration, because we assume that EWP are expected to produce responses to CI faster than expected, we define the initial training labels on the basis of *Centered log Response Time* (CT), we label the extremely fast responses as aberrant, and label the extremely slow responses as normal (the criterion for extreme speed is considered as a design factor in the experiment). In each subsequence iteration, although the aberrant responses identified in the finished iterations can be used to train the current classifier, these responses may not still adequately represent the remaining unlabeled data. To circumvent this problem, we still use the unlabeled responses that are extremely fast or slow as training data as more iterations are performed.

This training data preparation raises a concern. That is, although the responses containing preknowledge are supposed to have shorter response times than normal responses, the difference might be not significant in the later iterations of the classification procedure. To address this concern, we propose to use an index called *preknowledge propensity*, calculated from the aberrant responses identified so far, to describe the extent to which we believe a particular examinee/item has preknowledge. Before each iteration, this procedure updates the CT of each unlabeled response using the preknowledge propensities of the corresponding item and examinee to increase contrast.

Once the SVM classifier is trained, the self-training algorithm makes predictions for all unlabeled data in Step 1B. The SVM is a robust classification model, which attempts to place a classification boundary between the classes but as far as possible from the samples. However, due to the limited training data, the classifier may not be able to accurately label all the unlabeled responses, particularly in the early period of the classification procedure. To avoid labeling normal responses as aberrant, we use the responses with a high probability of being aberrant only as testing samples.

In step 1C, the self-training algorithm adds the unlabeled points with the most confidence to the labeled response set. Because we intend to recognize aberrant responses, we add only the responses classified as aberrant to the labeled response set, which is also the detected aberrant response set. After the classification procedure is completed, we output and use all intermediate aberrant response sets from all iterations for flagging suspicious examinees and items.

The stopping criterion in a typical self-training algorithm is that all the data points are labelled. In our classification procedure, we aim to set a stopping criterion that minimizes the number of responses under preknowledge in the unlabeled data. If unlabeled data responses are still present under preknowledge, as response times for EWP to CI are usually spuriously small, the observed variance would be larger than expected. Thus, we stop the classification process when the variance of response times among the remaining unlabeled responses is smaller than the expected variance of responses without preknowledge.

As for the features used in aberrant response classification, our choice is motivated by the belief that the response time reduction caused by preknowledge might be observed at examinee and item levels. For an examinee with preknowledge (a compromised item), their answers to CI (from EWP) are likely faster than they are for secure items (regular examinees). Thus, we use two features in the classification procedure: the *examinee-level centered log response times* (ECT) and the *item-level centered log response times* (TCT), which reflect the response speed after controlling for the examinee speed and time requirement for the item and examinee, respectively. As with the use of CT to select training samples, the caveat of using ECT and TCT to classify responses is that the difference between the two response classes might be not significant in later iterations of the classification procedure. Thus, before each iteration, we also update the ECT and TCT of each unlabeled response using the preknowledge propensities of the corresponding item and examinee to increase contrast.

Mathematical Details of the ML Algorithm

Pre-processing Response Time Data

The input data set consists of binary scores and log response times. Let x_{ij} denote the score of examinee i ($i = 1,2, \dots, I$) on item j ($j = 1,2, \dots, J$); $x_{ij} = 1$ indicates a correct response and $x_{ij} = 0$ an incorrect one; t_{ij} is the log response time of examinee i to item j.

The CT and classification features: ECT and TCT are computed by centering log response times. Let $\bar{t}_{i.}$ be the average log response time of examinee i over all administered items, and $\bar{t}_{.j}$ is the average log response time to item j over all examinees. Then the CT, ECT, and TCT for examinee i and item j are respectively computed as $CT_{ij} := t_{ij} - \bar{t}_{i.} - \bar{t}_{.j}$, $ECT_{ij} := t_{ij} - \bar{t}_{i.}$, and $TCT_{ij} := t_{ij} - \bar{t}_{.j}$.

Updating Centered Response Times for Classification

In the beginning of each iteration of Step 1, the CT, ECT and TCT are updated based on the CT, ECT and TCT of the previous iteration, respectively. The goal of the update is to decrease the CT, ECT and TCT of the aberrant responses to smaller values compared to the previous iteration, and thus to increase the contrast between the classes. However, the true classifications of responses are unknown. Thus, we calculate a preknowledge propensity for each examinee/item to quantify the extent to which we believe the examinee/item has preknowledge. Because the examinee/item having more aberrant responses implies a higher propensity of item preknowledge, the preknowledge propensity of examinee i (p_i^E) and of item j (p_j^T) are calculated by the percentage of aberrant responses among all responses of examinee i and item j, respectively. Suppose R^A is the current set of detected aberrant responses, L is the test length, and Ad_j is the number of administered times for item j. Then

$$p_i^E := \frac{|\{j|(i,j) \in R^A\}|}{L} \qquad and \quad p_j^T := \frac{|\{i|(i,j) \in R^A\}|}{Ad_j}. \tag{1}$$

A response with a higher preknowledge propensity at examinee or item level has a greater reduction in CT, ECT and TCT.

Because the amount of reduction should also consider the scale of CT, ECT and TCT, the standard deviation of CT, ECT and TCT are also used in the update. Suppose $SD_{i.}^{CT}$, $SD_{i.}^{ECT}$ and $SD_{i.}^{TCT}$ are the standard deviation of CT, ECT and TCT, respectively, among the unlabeled responses of examinee i in the previous iteration, and those for item j are $SD_{.j}^{CT}$, $SD_{.j}^{ECT}$ and $SD_{.j}^{TCT}$, respectively. Consequently, CT_{ij}, ECT_{ij}, and TCT_{ij} are updated as

$$CT_{ij} \leftarrow CT_{ij} - p_i^E \times SD_{i.}^{CT} - p_j^T \times SD_{.j}^{CT},$$

$$ECT_{ij} \leftarrow ECT_{ij} - p_i^E \times SD_{i.}^{ECT} - p_j^T \times SD_{.j}^{ECT},$$

$$TCT_{ij} \leftarrow TCT_{ij} - p_i^E \times SD_{i.}^{TCT} - p_j^T \times SD_{.j}^{TCT}.$$

Note that in the first iteration, since the current set of detected aberrant responses is empty, the update leads to no changes in CT, ECT and TCT.

Selecting Training Samples and Training a Classification Model

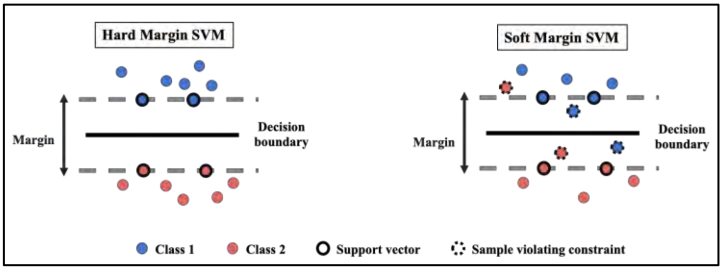
In this step, which is shown in the right panel of Figure 1, we first select several of the fastest responses as aberrant training samples ($R^{TR(A)}$), and several of the slowest responses as normal training samples ($R^{TR(N)}$) from the unlabeled response set (R^{U}), then train a SVM model that classifies responses as either normal or aberrant.

The number of selected fastest responses, as well as the number of selected slowest responses, is denoted as z. This training-sample selection criterion z is expected to be a small value, and will be considered as a design factor to explore its impact on detection. In the first self-training iteration, the set of unlabeled responses R^{U} is 60 % of the correct responses; otherwise, the R^{U} is the response set including all the sampled responses except the ones that have been identified as aberrant.

There are two types of SVM including hard-margin SVM, which requires all data points be classified correctly, and soft-margin SVM, which allows misclassification of outliers (Abu-Mostafa et al., 2012). Two examples are presented in Figure 2. As the boundary between normal and aberrant samples becomes more blurred in the later iterations, we employ soft-margin SVM. For soft-margin SVM, an error penalty parameter C has to be specified to describe the weight of the penalty term for misclassification. A large value of C makes soft-margin SVM similar to hard-margin SVM, while a small C might result in overfitting (Abu-Mostafa et al., 2012). We use a commonly used intermediate value, $C = 1$.

Figure 2

Example of classification using Hard-Margin SVM vs. Soft-Margin SVM.



In addition, SVM can perform a non-linear classification efficiently using kernel functions, such as polynomial kernel and radial basis function kernel (Abu-Mostafa et al., 2012). Three examples are presented in Figure 3. Since the reductions of response times in compromised responses are expected to be reflected at both the examinee and item level, the aberrant responses might be distributed in the bottom left corner of the feature space, forming an approximately fan-shaped area. An example is presented in Figure 4. Therefore, we use kernel SVM with a 3rd-degree polynomial kernel function.

Figure 3

Examples of SVM with Different Kernels.

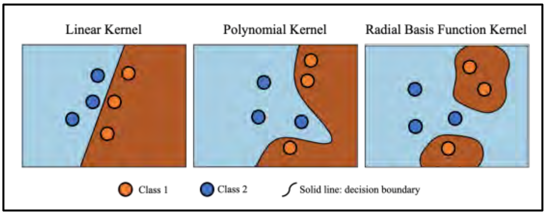
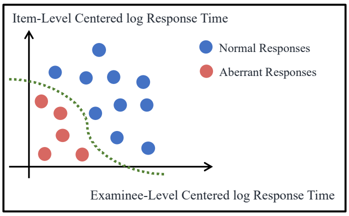


Figure 4

A Possible Distribution of Responses in the Feature Space.


Item-Level Centered log Response Time
Normal Responses
Aberrant Responses
Examinee-Level Centered log Response Time

Selecting Testing Samples and Conducting Classification

In this step, we first select testing samples (R^{TS}) from the unlabeled response set R^{U} used in the previous step. The responses whose corresponding items and examinees both appear in the aberrant training data are used as R^{TS}. Then we predict the classes of the R^{TS} using the SVM classifier trained in the previous step.

Suppose the examinees and items associated with the aberrant training responses are respectively denoted as $E^{\mathrm{TR}(A)} := \{i \mid \exists j\ (i,j) \in R^{TR(A)}\}$ and $T^{\mathrm{TR}(A)} := \{j \mid \exists\ i\ \ (i,j) \in R^{TR(A)}\}$. Then the testing response set is

$$R^{TS} := \{(i,j) \mid \ j \in \mathrm{T}^{\mathrm{TR}(A)} \ \text{ and } \ i \in E^{\mathrm{TR}(A)} \ \text{ and } \ (i,j) \in R^{U}\}.$$

Note that with this selection strategy, all the aberrant training samples are also used as testing samples, and their labels are set according to the classification result. This means their labels could switch from aberrant to normal or vice-versa.

Updating the Aberrant Response Set

After classification, the testing responses are predicted as either normal or aberrant. The responses predicted as aberrant are added to the aberrant response set R^{A} and are removed from the unlabeled data set R^{U}, namely,

$$R^{A} \leftarrow R^{A} \cup R^{TS(A)}$$

$$R^{U} \leftarrow R^{U} \backslash R^{TS(A)}.$$

Classification Iteration Stopping Criterion

The algorithm stops when the variance of response time among the remaining unlabeled responses is smaller than the expected variance of responses without preknowledge (var_{exp}). We estimate var_{exp} by the variance of response times among all the correct responses that are slower than the mode. The log response times of a data set that presents no preknowledge usually follow a normal distribution (e.g., van der Linden & Sotaridona, 2006); hence its variance can be estimated using only the slower half of the response times (e.g., the average Euclidean distance between the data points greater than the mean and the mean is the same as the variance of data), which are unlikely to be compromised. As the mode of the response time is less sensitive to deviations due to preknowledge than the mean and median, the slower half of the response times are defined as being slower than the response time mode.

Supposing that the mode of the response time among all the correct responses is t^{mo}, var_{exp} is calculated as:

$$var_{exp} = \frac{\sum_{(i,j)\in R^{SC}}\left(t_{ij} - t^{mo}\right)^2}{|R^{SC}|}, \tag{2}$$

where $R^{SC} = \{(i,j)|t_{ij} > t^{mo}, x_{ij} = 1\}$. Because the number of modes (each with frequency of 1) would be large if too many decimal places are kept, to find a representative value, we round t_{ij} to two decimal places for each response time when calculating t^{mo}.

In addition, to ensure that the algorithm works appropriately, the classification procedure stops when no response is predicted as aberrant, or the number of unlabeled responses is smaller than the training-sample selection criterion z.

Identifying Suspicious Examinees and Items

After the classification procedure is stopped, we use all the interim and final aberrant response sets, $\{R_1^A, R_2^A, \ldots, R_K^A\}$, where K is the number of iterations, to identify the suspicious examinees and items via an iterative process. For a particular R_k^A, we consider the examinees/items appearing in the set R_k^A as the candidates of suspicious examinees/items, and flag as suspicious candidates with a strong preknowledge signal. The suspicious candidates flagged in the last iteration are outputted as the result of the current repetition.

The process evaluates the preknowledge signal of each candidate in the set of the suspicious examinees by a proposed index β^E and the signal of each candidate in the set of the suspicious items by β^T, where β^E is a measure of the extent to which the examinee's responses match the pattern of being significantly faster on CI than for uncompromised items and β^T is a measure of the item's matching of the pattern of being significantly faster on EWP than for other examinees. The index β^E compares *the average correct response time* (AT) for a specific examinee on CI and the AT for the same examinee on all remaining items. As the CI are unknown, we use the suspicious items flagged in the previous iteration as an estimation of the CI. Suppose T_{k-1}^S is the set of suspicious items identified for the $k-1^{th}$ iteration, $\complement T_{k-1}^S$ is the complement of T_{k-1}^S, E_k^{CS} and T_k^{CS} are the sets of candidates of suspicious examinees and suspicious items in the k^{th} iteration,

$$E_k^{CS} = \{i|\exists j, (i,j) \in R_k^A\} \quad \text{and} \quad T_k^{CS} = \{j|\exists i, (i,j) \in R_k^A\},$$

and $AT_{M,N}$ is the AT for an examinee set M and an item set N,

$$AT_{M,N} = \overline{CT}_{\{(i,j)|i\in M, j\in N, x_{ij}=1\}} = \frac{\sum_{\{(i,j)|i\in M, j\in N, x_{ij}=1\}} CT_{ij}}{|\{(i,j)|i \in M, j \in N, x_{ij} = 1\}|}, \tag{3}$$

then β_i^E for examinee i in k^{th} iteration is calculated as:

$$\beta^{E}_{i(k)} = \begin{cases} AT_{i,\complement T^{S}_{k-1}} - AT_{i,T^{S}_{k-1}}, & k > 1 \\ AT_{i,\complement T^{CS}_{k}} - AT_{i,T^{CS}_{k}}, & k = 1 \end{cases}. \tag{4}$$

The index β^{T} is computed in a similar way, comparing the AT on a specific item for EWP and the AT on the same item for all remaining examinees. Since the EWP are unknown, E^{S}_{k}, the set of suspicious examinees flagged in this iteration, are used as an estimate of EWP. Suppose $\complement E^{S}_{k}$ is the complement of E^{S}_{k}, then β^{T}_{j} for item j in k^{th} iteration is calculated as:

$$\beta^{T}_{j(k)} = AT_{\complement E^{S}_{k},j} - AT_{E^{S}_{k},j}. \tag{5}$$

After β^{E} (β^{T}) is calculated for each candidate of the suspicious examinees (items), this approach flags the candidates having β^{E} or β^{T} greater than 0 as the suspicious. Thus, in the k^{th} iteration, the set of suspicious examinees is $E^{S}_{k} = \{i | \beta^{E}_{i(k)} > 0 \ and \ i \in E^{CS}_{k}\}$, and the set of suspicious items is $T^{S}_{k} = \{j | \beta^{T}_{j(k)} > 0 \text{ and } j \in T^{CS}_{k}\}$.The final values, E^{S}_{K} and T^{S}_{K}, are outputted as the result for the current repetition.

Repetition Stopping Criterion

Following the ensemble learning algorithm, we conduct repetitions continuously until the number of repetitions reach a set value. Because there is no golden rule for the number of repetitions, we consider this value as a design factor in the experiment to assess the approach's sensitivity to this value.

Combining Results of Repetitions

After all the repetitions are stopped, we combine the results from all repetitions to a final detection result following the strategy proposed in PW22. We first count the times of each examinee/item being flagged as suspicious, and then output the examinees/ items whose flagged times are no less than a criterion as the final detected examinees/items.

As with PW22, the examinee-criterion and item-criterion are determined empirically. Suppose we have set P possible examinee-criteria and Q possible item-criteria, firstly, each examinee-/item-criterion is used to generate an examinee/item set that includes all the examinees/items whose flagged times are no less than the criterion. Next, for each pair of generated examinee set and generated item set, we calculate a preknowledge signal index β that is a sum of β^{E} and β^{T}. Let the generated examinee sets be $E^{G}_{1}, E^{G}_{2}, \dots, E^{G}_{P}$, and the generated item sets be $T^{G}_{1}, T^{G}_{2}, \dots, T^{G}_{Q}$, then for the combination of a generated examinee set E^{G}_{p} ($p = 1,2 \dots,$) and a generated item set T^{G}_{q} ($q = 1,2, \dots, Q$), the preknowledge signal β_{pq} is calculated as

$$\beta_{pq} = \beta_p^E + \beta_q^T = \left(AT_{E_p^G,\mathrm{C}T_q^G} - AT_{E_p^G,T_q^G}\right) + \left(AT_{\mathrm{C}E_p^G,T_q^G} - AT_{E_p^G,T_q^G}\right). \qquad (6)$$

The criteria generating the combination with the greatest β are used as the examinee- and item- criteria, and the corresponding E^G and T^G are output as the final set of detected examinees and items.

We experiment with examinee-criteria threshold values of 80 %, 82 %, 84 %, 86 %, 88 %, 90 %, 92 %, 94 %, 96 %, 98 %, and 100 % of the maximum flagged times across all examinees, and similarly, with item-criteria threshold values of 80 %, 82 %, 84 %, 86 %, 88 %, 90 %, 92 %, 94 %, 96 %, 98 %, and 100 % of the maximum flagged times across all items.

Confidence Score Corresponding to the Detection Results in CAT

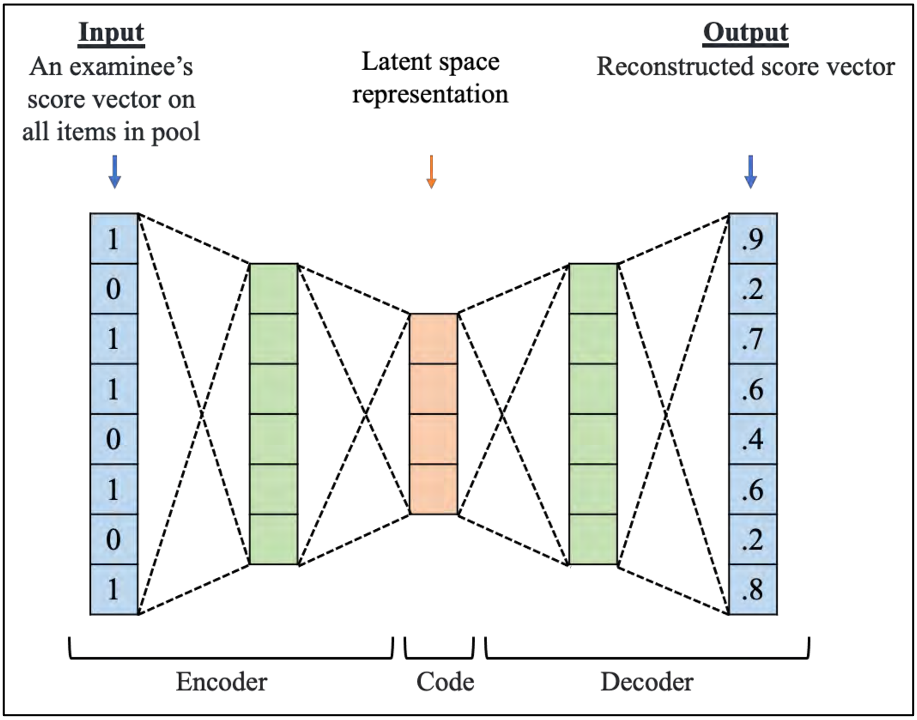
As mentioned above, we assume that item preknowledge will manifest as EWP, producing responses to CI more accurate than expected on average. Based on this assumption, we adapt the confidence score proposed in PW22 to provide a measure of confidence that the detection result corresponds actual item preknowledge. The confidence score evaluates the extent to whichthe scores of detected examinees/items are unexpectedly high, and was calculated in two steps: (1) the expected scores were predicted by an autoencoder; (2) the degrees of un-expectancy at the examinee and item level were calculated by the expected scores; the confidence score was computed as the average of these two un-expectancy degrees.

Our confidence score differs from the confidence score in PW22 in two ways. First, PW22 directly uses the score data as the training/testing data in the first step. However, the autoencoder cannot be trained by or applied to sparse matrices as are typical with CAT response data. Thus, we proposed a strategy to estimate missing values and use the data after imputation as the training/testing data. Second, the confidence score in PW22 considers the un-expectancy degree at both examinee and item levels, while our proposed confidence score is calculated by the un-expectancy degree at the examinee level only. PW22 shows that the un-expectancy degree at the item level makes the confidence score ineffective (e.g., the detections with low error rates have low confidence scores) when there are multiple groups of EWP having access to multiple sets of CI and these sets are only partially overlapped. As the examinees will likely be assigned to different items in CAT, the EWP might be exposed to different CI, and thus the confidence score in PW22 might be ineffective for detection results in CAT. To overcome this problem, because the ML approach detects EWP and CI simultaneously and better performance in the detection of EWP usually implies better performance in the detection of CI, we use the un-expectancy degree at the examinee level only as the confidence score.

Predicting Expected Responses by Autoencoder

5

An Example of the Autoencoder.


Input
An examinee's score vector on all items in pool
Latent space representation
Output
Reconstructed score vector
1
0
1
1
0
1
0
1
.9
.2
.7
.6
.4
.6
.2
.8
Encoder
Code
Decoder

In this step, following the PW22 method, we first train an autoencoder and then use the trained model to predict expected scores. The autoencoder is used because it performs well in data denoising (Vincent, Larochelle, Bengio, & Manzagol, 2008). This model is a fully connected neural network, which first compresses the input into a reduced representation, and then generates an output from the reduced that is as close to the original input as possible (an example is presented in Figure 5). Since the goal is to reconstruct the input, the removed information should be the noise in the input, and the reduced representation as well as the reconstructed output should keep the essential information in the input. Because the predicted expected scores are expected to reflect the examinee's true ability and ignore the noise in the input scores, we use the autoencoder to predict expected scores.

The architecture of our autoencoder is the same as the one in the PW22 method. The input is an examinee's response score vector to all the items in the item pool. To compress the input by 50%, the sizes of the hidden layers are 100%, 90%, 80%, 70%, 60%, 50%, 60%, 70%, 80%, 90%, and 100% of the test length. The output is the reconstructed score vector. For a detailed description of the autoencoder, refer to PW22.

We prepare the training/testing data for the autoencoder in a way different from PW22. The training data is prepared based on the observed binary scores. Firstly, we impute the scores of missing responses with estimated binary scores. Secondly, to make sure the training data reflect the uncompromised response patterns as much as possible, we replace the observed scores with estimated binary scores for the responses of the detected examinees to the detected items. To estimate the binary score for a particular response, we use a Bernoulli trial with a probability, which is computed using the average scores of the examinee and of the item. When calculating the average scores, to reduce the impact of item preknowledge, we use only those responses believed to be uncompromised (i.e., excluding responses from detected examinees to detected items). Suppose X' is the binary score matrix that does not contain the detected examinees' scores on the detected items, then the estimated binary score for the response from examinee i to item j is a draw from a Bernoulli trial with $p = \frac{\overline{x'}_{i.}+\overline{x'}_{.j}}{2}$, where $\overline{x'}_{i.}$ and $\overline{x'}_{.j}$ are the average scores respectively of examinee i and item j, both calculated using X'.

The testing data are also prepared based on the observed binary scores. However, only the missing scores are replaced with estimated binary scores. After the resulting matrix is input to the model, the reconstructed scores are treated as the expected scores.

Calculating Confidence Scores

We use the degree of un-expectancy at the examinee level only to calculate the confidence score. In each detection, the suspicious examinees and items are flagged based on the same aberrant response set. If the aberrant response set corresponds to item preknowledge, both the flagged examinees and items are likely associated with true EWP and CI. Thus, to some extent, better performance in detecting EWP implies better performance in detecting CI, and the un-expectancy degree at the examinee level can be used as the confidence score.

We calculate the degree of un-expectancy at the examinee level in the same way as PW22. A response is treated as unexpectedly correct if it has an expected score lower than 0.5 but is observed to be correct. Although the proportion of unexpectedly correct responses among all the responses from the detected examinees could be used to evaluate the degree of un-expectancy, this proportion might be an over- or under-evaluation when the prediction of expected scores is not accurate. Because such over-/under-evaluation also happens among the responses of the remaining examinees, the examinee-level un-expectancy degree is designed to be a ratio. This ratio compares the proportion of unexpectedly correct responses among all the responses from the detected examinees to the one among the remaining responses. A confidence score close to 1 or smaller than 1 indicates that the detection result does not reflect actual item preknowledge. We set the criterion for identifying low confidence score at 1.15.

Simulation Study

To evaluate the performance of the new item-preknowledge detection approach and the confidence score, we performed a simulation study in which we (a) simulated response data that includes item preknowledge and rapid guessing; (b) applied the ML algorithm and confidence score to detect EWP and CI; and (c) analyzed the performance of the ML algorithm and the confidence score under various conditions.

We simulated responses in a CAT framework, while applying different models to generate responses contaminated by item preknowledge. A common criticism of data-driven detection methods is that they may be sensitive to other forms of aberrance. For item preknowledge, rapid guessing may have the greatest negative impact on detection performance, since it also leads to fast responses. Thus, to demonstrate whether the ML approach can effectively distinguish item preknowledge from other anomalous behaviors in detecting preknowledge, we also simulated rapid guessing. In doing so, we manipulated four design factors: the proportion of CI (.1, .2, .4), the proportion of EWP (.1, .2, .4), the training-sample selection criterion (250, 500) and the number of repetitions (30, 60). The test length, the item pool size and the number of examinees were fixed at 50, 500 and 1000. Crossing all factors resulted in 36 $(3 \times 3 \ \times 2 \ \times 2)$ study conditions. In addition, a baseline condition that involved no preknowledge was also considered, adding 4 $(1 \times 1 \times 2 \times 2)$ more study conditions. For each combination of the proportions of CI and EWP, 30 replications were simulated; CI/EWP were randomly sampled from all items/examinees without replacement for each replication separately. Thus, a total of 300 $(3 \times 3 \ \times 30 + 1 \times 1 \ \times 30)$ datasets were simulated. For each dataset, we conducted detection twice, setting the training-sample selection criterion to 250 and 500, respectively. For each detection, the detected examinees and items were outputted when the number of repetitions reaches 30 and 60 for the purpose of performance evaluation.

The simulation was implemented in Python 3.7. The source code is included with this paper as supplemental material.

Data Simulation Models

Generating Uncompromised Responses

We generated uncompromised responses and response times using the joint model for item scores and response times within the hierarchical framework of van der Linden (2007).

Level 1 models. In the first level, the three-parameter logistic (3PL) model was used to generate item responses as follows:

$$P(x_{ij} = 1|\theta_i, a_j, b_j, c_j) = c_j + \frac{1 - c_j}{1 + e^{-a_j(\theta_i - b_j)}}$$

where θ_i is the ability of examinee i, a_j, b_j and c_j are the discrimination parameter, difficulty parameter and guessing parameter of item j, respectively, and $P(x_{ij} = 1)$ is the probability of examinee i giving a correct response to item j.

The response times were generated from the log-normal response time model (van der Linden & Sotaridona, 2006), as follows:

$$t_{ij} \sim N(\phi_j - \zeta_i, 1/\lambda_j^2),$$

where ζ_i denotes the speed parameter of examinee i, λ_j and ϕ_j are the discrimination parameter and time-intensity parameter of item j, respectively, and t_{ij} is the log response time of examinee i on item j.

Level 2 model. In the second level, the person parameters were generated from a bivariate normal distribution as:

$$(\theta_i, \zeta_i) \sim N_2(\mu_P, \Sigma_P),$$

with

$$\mu_P = (\mu_\theta, \mu_\zeta),$$

$$\Sigma_P = \begin{pmatrix} \sigma_\theta^2 & \sigma_{\theta,\zeta} \\ \sigma_{\zeta,\theta} & \sigma_\zeta^2 \end{pmatrix}.$$

Following Choe et al. (2018), we used the following values of μ_P and Σ_P:

$$\mu_P = (0.5, 0),$$

$$\Sigma_P = \begin{pmatrix} 1 & 0.12 \\ 0.12 & 0.02 \end{pmatrix}.$$

For purposes of this simulation, all model parameters and hyperparameters were the same as were used by Choe et al. (2018) in their simulation, and were based on the application of the two-level hierarchical framework to real data from a high- stakes, large-scale standardized CAT. This real data consists of raw responses and response times from approximately 2000 examinees with an item pool of 540 items. Here, the size of the item pool was fixed to be 500; accordingly, in each replication, 500 items were randomly selected from the 540 used in Choe et al. The descriptive statistics of the item parameters are presented in Table 1.

Table 1

Descriptive Statistics for Parameters of 540 Items

	a	b	c	λ	ϕ
Mean	0.836	0.319	0.161	1.757	0.366
SD	0.276	1.112	0.057	0.522	0.500
Max	1.840	3.011	0.500	3.186	1.802
Min	0.247	-4.067	0.000	0.716	-0.890

Generating Responses Containing Item Preknowledge

The EWP and CI were determined by the particular simulation condition. For the responses of EWP to CI, the item scores were generated randomly from a Bernoulli distribution with $p = .9$, irrespective of the item's difficulty or the examinee's θ (Eckerly, 2017; Sinharay, 2017). With this probability, the item responses corresponding to item preknowledge had an expected accuracy of 90 %. As in Choe et al. (2018), the log response times were obtained by drawing values randomly from a normal distribution, $N(-2, 1/3.5^2)$. Using this distribution, we simulated a response time that is objectively fast, irrespective of the examinee's ζ.

Generating Responses Containing Rapid Guessing

A common pattern of rapid guessing is that the examinee produces a sequence of rapid guesses to multiple items towards the end of the test as time expires in the hope of getting some correct by lucky guessing (Wise, 2017). Rapid guessing was also simulated, manifesting as several examinees giving rapid guesses to the last several items on the test. The examinees with rapid guessing were assigned by randomly sampling 10 % of the normal examinees without replacement. For each such examinee, we used the $\text{Gamma}\,(10, .5)$ distribution to determine the number of guessed answers, producing an average of 5 rapidly guessed answers (out of 50).

For the selected rapid-guessing responses, the item scores were generated by drawing values randomly from a Bernoulli trial with $p = .25$, because four is the most common number of answer options for multiple-choice items. Paralleling what was done for responses contaminated with preknowledge, the log response times for these responses were simulated randomly from the normal distribution $N(-2, 1/3.5^2)$.

CAT Simulation Algorithm

The CAT algorithm used in this paper employed the a-stratification with b-blocking item selection approach (Chang, Qian, & Ying, 2001). Before using the CAT algorithm to simulate responses, we (1) prepared item and examinee parameters according to the methods mentioned above, (2) stratified items by b-parameter values, and then grouped them by a-parameter values from each b-parameter stratum to construct several item strata; the number of item strata was fixed to be 5 and the item pool size was fixed to be 500; the item pool was divided into 5 strata of 100 items each; (3) specified EWP and CI by random selection with the number of such examinees/items determined by the particular simulation condition. Then we iterated the following three steps for each examinee to generate responses.

1. Ability Estimation. The updated ability, $\hat{\theta}$, was estimated in this step. At the beginning of the test, as there was no response data for ability estimation, we set the updated ability equal to drawing a random value from normal distribution $N(0,1)$. After response data was collected from one or more items, $\hat{\theta}$ was updated by the expected a posteriori (EAP) estimate of ability.

2. Item Selection. Suppose the number of administered items was n_{adm} and the number of items that would be administered among each stratum was n_{s} (as the test length and the number of item strata were fixed at 50 and 5, n_s was 10). Then this item selection approach selected as the next item the one with b-parameter closest to $\hat{\theta}$ in the $(\lfloor n_{\mathrm{adm}}/n_{\mathrm{s}} \rfloor + 1)^{th}$ item stratum.

3. Generation of Item Responses. When the current examinee was an examinee with preknowledge and the selected item was a compromised item, we generated the response using the method for simulating compromised responses. If the current response was selected to have rapid guessing, it was generated using the corresponding method. Otherwise, the response was considered a typical/non-aberrant response and was simulated using the joint model of van der Linden (2007). As a result of this strategy, the simulated contamination affected both the estimation of θ and the item selection (e.g., simulating a compromised response for an examinee with preknowledge likely increased the interim θ of this examinee, which likely resulted in the administration of an item that was different from what this examinee would have seen without preknowledge).

Results

We first evaluated the performance of the ML approach, and then assessed the effectiveness of the confidence score.

Performance of the ML Approach

The performance of the ML approach was measured using three metrics: the false-negative rate (FNR), false-positive rate (FPR), and precision. False negatives refer to the situation in which an examinee/item is associated with EWP/CI, but is not detected by the ML approach. Thus, the FNR is calculated as the ratio of the number of true EWP/CI that are not detected to the total number of true EWP/CI. False positives, on the other hand, occur when an examinee/item is not associated with EWP/CI but is detected as such. Hence, the FPR is computed as the ratio of the number of examinees/items that are incorrectly identified as EWP/CI to the total number of flagged examinees/items. The precision is the proportion of correctly identified examinees/items to the number of detected examinees/items.

In addition, to understand how sensitive the ML approach is to the specific levels of various design factors, we conducted an analysis of variance (ANOVA) for the results on the detection of EWP and CI. Since the data could not meet the assumptions of ANOVA, Aligned Rank Transform (ART) ANOVA, which is a non-parametric approach to ANOVA, was used instead. Each ART ANOVA used four simulating factors as independent variables, and the FNR, FPR and precision as dependent variables. The ART ANOVA results and corresponding effect sizes (partial η^2) were attached in Appendix A. Also, to investigate the consistency of performance in detection of EWP and CI, we calculated the standard deviation across 30 replications for each of the evaluation metrics. The results were attached in Appendix B.

Moreover, to compare the ML approach with the existing approaches, we applied the approach proposed in Boughton, Smith, and Ren (2017; BSR) to detect EWP and CI in the same simulated data. According to BSR, we first estimated the posterior predictive distribution of log response time based on a lognormal response time model defined by van der Linden (2007), then standardized each log response time using the mean and variance of the predictive distribution, and lastly flagged examinees and items using the standardized response times.

Results on Detection of EWP. The ART ANOVA results for the detection of EWP show the proportions of EWP and CI have a great impact on the detection performance. When the dependent variable is the FPR (or precision), although all four factors have statistically significant main effects, the proportion of EWP, the proportion of CI and the interaction have the largest effect sizes, the partial η^2s of which are .10 (.09), .12 (.08) and .10 (.07). For FNR, three factors (the proportions of EWP and CI, and the number of repetitions) have statistically significant main effects. The effect sizes corresponding to the proportion of EWP (partial $\eta^2 = .31$), the proportion of CI (partial $\eta^2 = .58$), and their interaction (partial $\eta^2 = .22$) are considerably larger than that for the other factors (partial η^2s $< .03$).

Next, we present some details regarding the performance in the detection of EWP. Because the effect sizes for both the training-sample selection criterion and the number of repetitions are so small, results are shown for only one level of each (when the training-sample selection criterion is 500 and the number of repetitions is 60). Results for the other conditions are available from the authors upon request.

Figure 6 shows the FNR, FPR, and precision for the detection of EWP. Note that in the interest of clearly seeing how results vary across conditions, the vertical scale of the FPR graph extends from 0 to .10, whereas the other two graphs show results from 0 to 1.0. In general, the ML approach performs well in detecting EWP. The FNR decreases as the proportion of EWP or CI decreases. Although it is relatively high when both the proportions of CI and EWP are low, it is still acceptable. The ML approach controls the FPR at a very low level. When the examinees have preknowledge, the FPR is smaller than .02 under most conditions. Also, the ML approach produces very high precisions across all the conditions, indicating that most of the flagged examinees are truly EWP. Meanwhile, according to the standard variance of the evaluation metrics, the approach has stable performance in detecting EWP.

Figure 6

Performance for the ML Approach in the Detection of EWP

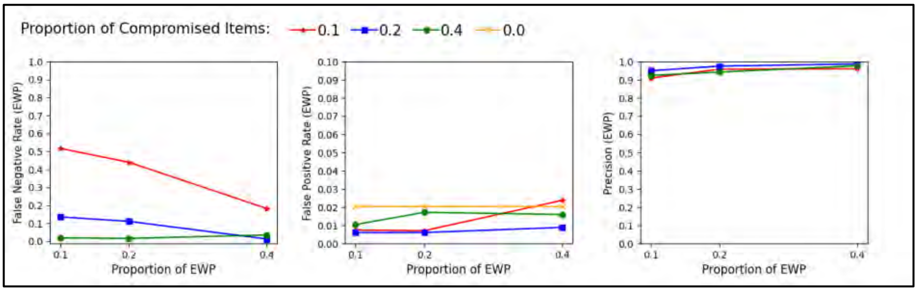
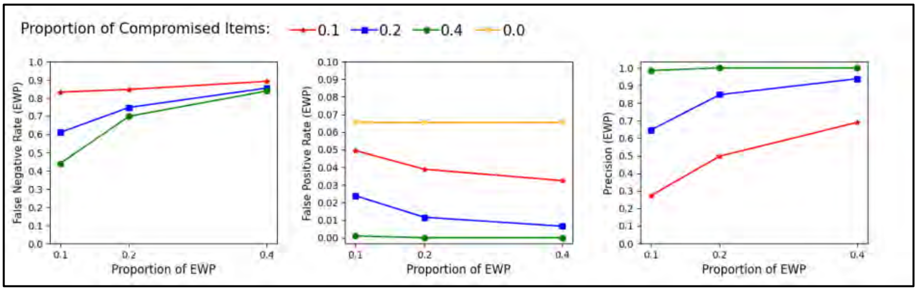


Figure 7 shows the performance of BSR in the detection of EWP. Compared to BSR, the ML approach generates much lower FNRs under all conditions, and lower FPRs and higher precisions under most conditions. The one exception is when the proportion of CI is .4. In this condition, the FPR for BSR is 0; hence, the precision is 1 across all replications. However, the primary objective of the methodology is to identify true preknowledge while limiting the FPR to an acceptable level, and under those same conditions, FNR for BSR is much higher than that for the ML approach developed here, while FPR for BSR is overly conservative. Thus, compared to BSR, the ML approach significantly improves the overall performance in the detection of EWP.

Figure 7

Performance for BSR in the Detection of EWP



Results on Detection of CI. The ART ANOVA results for the detection of CI are similar to those for the detection of EWP. When the dependent variable is the FPR (or precision), although all four factors have statistically significant main effects, the proportions of EWP and CI have the largest effect sizes, the partial η^2s of which are .15 (.27) and .30 (.77). For FNR, three factors (the proportions of EWP and CI, and the number of repetitions) have statistically significant main effects. The effect sizes corresponding to the proportion of EWP (partial η^2 = .48) is considerably larger than that for the other factors (partial η^2s < .04). Thus, similar to the detection of EWP, results for the remaining analyses are only presented for the simulation conditions where the training-sample selection criterion is 500 and the number of repetitions is 60.

Figure 8

Performance for the ML Approach in the Detection of CI

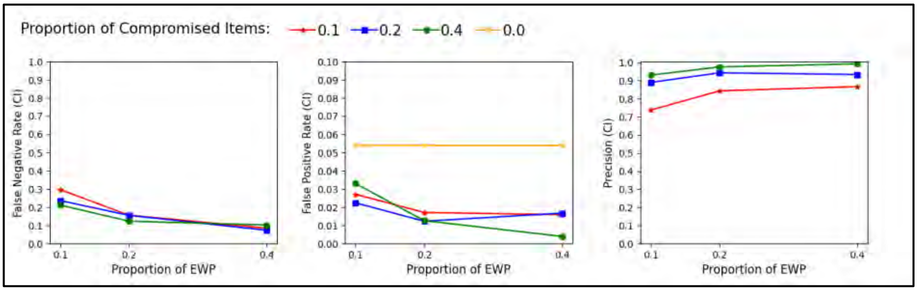


Figure 9

Performance for BSR in the Detection of CI

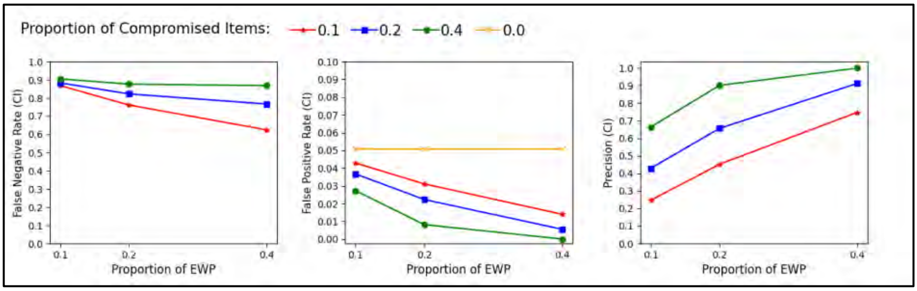


Figure 8 shows the FPR, FNR, and precision for the detection of CI. In general, the ML approach performs well in detecting CI. The FNR is at its highest when the proportion of EWP is low (e.g., .1), though in absolute terms, the ML approach produces FNRs smaller than .3 under all conditions. The FPR is generally small across all the conditions, increasing as the magnitude of item preknowledge decreases. The FPR appears to be relatively high when the data involves no item compromise. The precision is above .7 across all conditions and increases significantly as the proportion of CI increases. As for the standard variance, the performance in detecting CI across replications also shows high consistency.

Figure 9 shows the performance of BSR in the detection of CI. Overall, the performance of BSR in detection CI is inferior to that of the ML approach proposed here. Only when no CI is present, BSR produces slightly lower FPRs. However, the difference is negligibly small.

Figure 10

Performance for the ML Approach after the Confidence Score is Considered in the Detection of EWP

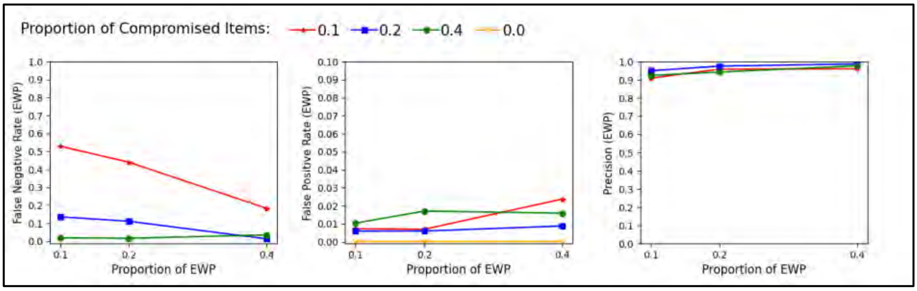
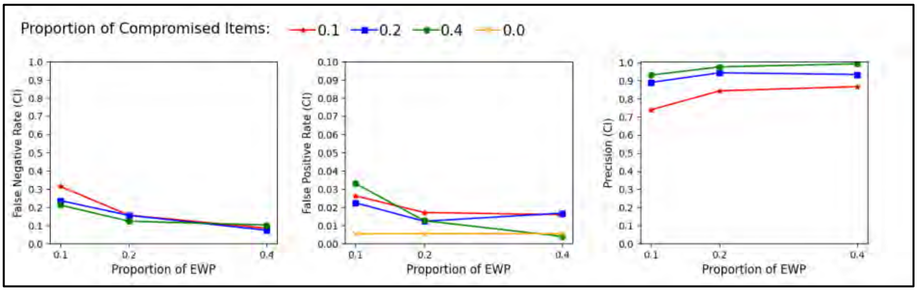


Figure 11

Performance for the ML Approach after the Confidence Score is Considered in the Detection of CI



Effectiveness of the Confidence Score

The confidence score is proposed to assist practitioners in deciding whether to use the detection result. To assess the effectiveness of our confidence score, we re-evaluated the data, treating any detected items or examinees as having no preknowledge, when the confidence score was smaller than the criterion. These adjusted results were then compared to the results before imposing the confidence score criterion to examine the overall impact.

Figures 10 and 11 present the data from Figures 6 and 8, respectively, after first imposing the confidence score criterion. For the detection of EWP, the performance improves after the confidence score is considered. The most important difference is with respect to method's performance in purely no-preknowledge data, where the FPR decreases from .02 to.0006. In the conditions for which preknowledge was simulated, the overall pattern of results changed very little. Similarly, for the detection of CI, the confidence score also helps to decrease the FPR for the no-preknowledge condition from .05 to .006. These results are consistent with PW22, indicating that the confidence score can improve detection performance, particularly under the no-preknowledge condition. Despite the success at reducing the FPR in the no-preknowledge condition, across true preknowledge conditions, results are effectively unchanged.

To obtain a better understanding of the effectiveness of the confidence score, we also created scatter plots to visualize the relationship between the FNR, FPR and confidence score under various simulation conditions for the detection of EWP and CI. These scatter plots were presented in Appendix C. Each dot in the scatter plots represents one result on the detection of CI or EWP. The position of each dot on the horizontal axis indicates the FNR, and the position on the vertical axis indicates the FPR. The color of dot demonstrates the confidence. Note that the horizontal scale is not presented in the first plot, as the FPR does not exist under the no-preknowledge condition. Consistent with the results shown in PW22, the confidence score is higher when the detection result is better.

Discussion

We propose a ML-based approach to simultaneously detect EWP and CI in CAT. To assist practitioners in deciding whether to use the detection result, the ML approach also provides a confidence score that represents the confidence that the detection result reflects actual item preknowledge. Simulation studies show that the ML approach performs well at flagging EWP and CI while limiting the erroneous detection to a very low level. The results also indicate that the confidence score is highly effective in providing helpful information for practitioners, particularly for the no-preknowledge condition.

The simulations show that the detection performance becomes better as the magnitude of preknowledge increases. The ML approach generates relatively high FPRs or FNRs when the magnitude of preknowledge is low. When the proportion of CI is low (e.g., .1), the detection of EWP generates relatively high FNRs. One explanation is that the ML approach flags as suspicious the examinees matching the pattern of being significantly faster on the detected items than on the remaining items (in Step *Identifying Suspicious Examinees and Items*). However, even though an examinee is truly associated with EWP, the preknowledge signal of this examinee might not be strong enough to be flagged when there are only a few CI. For the same reason, the detection of CI also generates relatively high FNRs, when the proportion of EWP is low (e.g., .1). According to this explanation, the FNR could be controlled at a lower level by adjusting the threshold of the preknowledge signal. Namely, when the number of detected items/examinees is small, the threshold is modified to a smaller value in the flagging of suspicious examinees/items.

Although the FPRs are consistent with where we would like them for typical experimental research studies in the educational and psychological sciences (e.g., $< .05$), given the costs associated with item development and the risks associated with falsely accusing examinees of cheating, when doing test security work, programs often like to see FPR that are lower. The FPR could be reduced by adjusting the possible examinee- and item- criteria in Step *Combining Repetition Results*, such as increasing the criteria to 90%, 92%, …, 100 % of the maximum flagged times. Also, in Step *Identifying Suspicious Examinees and Items*, modifying the threshold of the preknowledge signal in the other direction, changing it to a greater value, would help to control the FPR at a lower level.

When the magnitude of preknowledge is null, that is, the data involves no item preknowledge, the ML approach generates relatively high FPRs. This is not out of expectation. The ML approach will always flag a set of objects even in a no-preknowledge condition, as our fundamental argument is that the correct responses with relatively short response times among the population are aberrant. There are always responses with relatively short response times, and thus there are always flagged objects. Although the ML approach cannot recognize the no-preknowledge condition, the confidence score can successfully recognize this condition in most replications.

Compared to the confidence score in PW22, our confidence score has a better performance as it increases FNR very little. In PW22, most detections whose FNR increases after the confidence score was considered were for the case where the CI exposed to different EWP were only partially overlapped. Thus, one explanation for this improvement is that the nature of the ML approach allows us to calculate the confidence score using the un-expectancy degree at the examinee level only, so that our confidence score is still effective under the CI-partially-overlapped condition.

In addition to the promising detection performance, another notable contribution of the ML approach is that it provides an example of using a supervised learning model to classify responses when the compromised status is unknown. Most of the detection approaches that require no prior information are developed based on the response similarity or the consistency between the response patterns among examinees. However, these approaches could not work well when EWP have access to different subsets of CI or the testing form is CAT. To overcome this limitation, we use domain knowledge in preknowledge to label a small set of responses and develop a supervised classification algorithm based on the labeling result. To improve classification performance, we further integrate multiple techniques into the algorithm, such as semi-supervised learning and ensemble learning. Classification performance at an acceptable level is provided when all the strategies and techniques work together.

The ML approach described here makes choices for multiple criteria and thresholds (e.g., the criterion for identifying small confidence scores), which can be regarded as hyperparameters. Hyperparameters are parameters that can control the detection process. In practice, the nature of preknowledge problem varies. The practitioners can alter the hyperparameters in accordance with the nature of their problem to obtain better detection. The optimal selection of the hyperparameter values depends on the preknowledge situation. For example, when the proportion of responses with item preknowledge is high, using a relatively small threshold in combining the detection should produce better detection performance. Further, the optimal choice of the hyperparameter values is on the basis of the rigor of the detection. For instance, for a detection that is committed to controlling the false positive rate at an extremely low level, even at the expense of the false negative rate to some extent, selecting a greater value as the criterion for identifying a low confidence score should yield a more desired detection result. Moreover, in the application of confidence score, the structure of the autoencoder could be adjusted according to the complexity of the data. If the complexity is high, the users could increase the number of hidden layers and the dimension of the reduced representation to provide a more accurate prediction for the expected scores. If the user-defined hyperparameters deviate from the settings here, it is generally recommended that they assess the effectiveness of user-defined criteria using simulation studies that replicate the characteristics of the testing program and expected preknowledge condition to the greatest degree.

A limitation of this research is that only one model and one anomalous behavior were considered in the data simulation. Firstly, only the 3PL model was used to generate item responses. Because the detection does not depend much on response accuracy and does not involve the item parameters, the choice of the IRT model likely has little impact on the performance. However, there are many certification and licensure programs that use the Rasch or 2PL model. Hence, other IRT models could have been used to assess the detection performance of the ML approach under various conditions. Secondly, the lognormal distribution used to generate the response times involving preknowledge is a common and seemingly reasonable choice for the purposes of this study (Choe et al., 2018), but other distributions could have been used. Thirdly, although our approach was able to maintain strong psychometric properties even with random guessing simulated, the robustness of the ML approach to other types of aberrance should still be explored.

Another limitation is that, to calculate the confidence score, the ML approach imputes the missing responses using a single-imputation method based on average scores. While the single imputation is able to calculate an informative confidence score, other imputation methods should be explored. For example, the missing values could be predicted in the context of IRT modeling (e.g., Van Ginkel, Sijtsma, van der Ark, & Vermunt, 2010). Specifically, first, item and ability parameters are estimated using observed responses; second, the probability of correct response is calculated for each missing based on the estimated parameters; finally, the predicted score is a random draw from a Bernoulli distribution with the estimated correct probability. Also, the missing values could be predicted by multiple imputation (Rubin, 1987). The comparison of different imputation methods could be conducted by future studies.

There are several other limitations. First, the maximum amount of time examinees can spend is not limited in the simulation. In particular, there is likely no relationship between who is selected for rapid responses and the amount of time those examinees spent on other items. Simulating data in a way that is more reflective of the real situation can help provide more convincing evidence for the detection performance of ML methods (or, quite frankly, for any simulated research on the use of response time). Second, the proposed confidence score is designed as an omnibus test for the entire dataset. It would be helpful to develop a confidence score that indicates the degree of confidence that each detected examinee or item corresponds to item preknowledge. Third, the ML approach is based on ML and does not involve item response theory models or response times models in any manner. Integrating those models into an extended version of the ML algorithm might improve detection performance. Fourth, the detection uses dichotomous scores. However, examinees may receive partial credits in the form of ordinal scores rather than binary scores. The ML approach would be more applicable if partial credits can be handled. Finally, we assumed that EWP show a significant decrease in response time for CI. However, EWP may intentionally answer CI at normal speed. The detection performance would be greatly improved if this approach is extended to be able to recognize this situation. Exploring these avenues is left for future study.

References

Abu-Mostafa, Y. S., Magdon-Ismail, M., & Lin, H. T. (2012). Support vector machines. In *Learning from Data: A Short Course*. AMLBook.

Alpaydin, E. (2004). *Introduction to Machine Learning Cambridge*. Cambridge, MA: MIT Press.

Belov, D. I., and Armstrong, R. D. (2011). Distributions of the kullback leibler divergence with applications. *British Journal of Mathematical and Statistical Psychology, 64*(2), 291-309.

Boughton, K. A., Smith, J., & Ren, H. (2017). Using response time data to detect compromised items and/or people. In G. J. Cizek & J. A. Wollack, (Eds.), *Handbook of quantitative methods for detecting cheating on tests* (pp. 177–190). New York, NY: Routledge.

Chang, H. H., Qian, J., & Ying, Z. (2001). a-Stratified multistage computerized adaptive testing with b blocking. *Applied Psychological Measurement, 25*(4), 333-341.

Choe, E. M., Zhang, J., & Chang, H. H. (2018). Sequential detection of compromised items using response times in computerized adaptive testing. *Psychometrika, 83*(3), 650-673.

Cizek, G. J., & Wollack, J. A. (Eds.). (2017). *Handbook of quantitative methods for detecting cheating on tests.* New York, NY: Routledge.

Domingue, B. W., Kanopka, K., Stenhaug, B., Soland, J., Kuhfeld, M., Wise, S., & Piech, C. (2021). Variation in respondent speed and its implications: Evidence from an adaptive testing scenario. *Journal of Educational Measurement, 58*(3), 335-363.

Drasgow, F., Levine, M. V., and Williams, E. A. (1985). Appropriateness measurement with polychotomous item response models and standardized indices. *British Journal of Mathematical and Statistical Psychology, 38*(1), 67–86.

Eckerly, C. (2017). Detecting preknowledge and item compromise. In. G. J. Cizek & J. A. Wollack, (Eds.), *Handbook of Quantitative Methods for Detecting Cheating on Tests* (pp. 101-123). New York, NY: Routledge.

Figueiredo, M. A. T., & Jain, A. K. (2002). Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(3), 381-396.

Foster, D. (2013). Security issues in technology-based testing. In J. A. Wollack & J. J. Fremer (Eds.), *Handbook of Test Security* (pp.261-283). New York, NY: Routledge.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Autoencoders. In *Deep learning* (pp. 499-523). Cambridge, MA: MIT press.

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, *160*(1), 3-24.

Längkvist, M., Karlsson, L., & Loutfi, A. (2014). A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters, 42,* 11-24.

Liu, C., Han, K. T., & Li, J. (2019). Compromised item detection for computerized adaptive testing. *Frontiers in psychology*, *10*, 829.

Man, K., Harring, J. R., & Sinharay, S. (2019). Use of data mining methods to detect test fraud. *Journal of Educational Measurement, 56*(2), 251-279.

McLeod, L., Lewis, C., & Thissen, D. (2003). A Bayesian Method for the Detection of Item Preknowledge in Computerized Adaptive Testing. *Applied Psychological Measurement, 27*(2), 121–137.

Pan, Y., & Wollack, J. A. (2021). An unsupervised-learning-based approach to compromised items detection. *Journal of Educational Measurement, 58*(3), 413-433.

Pan, Y., & Wollack, J. A. (2022). An Ensemble-Unsupervised-Learning-Based Approach for the Simultaneous Detection of Preknowledge in Examinees and Items when Both are Unknown. https://doi.org/10.31234/osf.io/jtr78.

Rubin, D. B. (1987). *Multiple imputation for nonresponse in surveys*. New York, NY: Wiley.

Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8*(4), e1249.

Sinharay, S. (2017). Detection of item preknowledge using likelihood ratio test and score test. *Journal of Educational and Behavioral Statistics, 42*(1), 46–68.

Sinharay, S., & van Rijn, P. W. (2020). Assessing fit of the lognormal model for response times. *Journal of Educational and Behavioral Statistics. 45*(5), 534-568.

Suykens, J. A., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters, 9*(3), 293-300.

Thomas, S. L. (2016). *So happy together? Combining Rasch and item response theory model estimates with support vector machines to detect test fraud*. [Unpublished doctoral dissertation]. University of Virginia.

van der Linden, W. J. (2007). A hierarchical framework for modeling speed and accuracy on test items. *Psychometrika*, *72*(3), 287.

van der Linden, W. J., & Sotaridona, L. (2006). Detecting answer copying when the regular response process follows a known response model. *Journal of Educational and Behavioral Statistics, 31*(3), 283–304.

Van Ginkel, J. R., Sijtsma, K., van der Ark, L. A., & Vermunt, J. K. (2010). Incidence of missing item scores in personality measurement, and simple item score imputation. *Methodology, 6*(1), 17-30.

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. (2008, July). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning* (pp. 1096-1103). New York, NY: Association for Computing Machinery.

Wang, C., Xu, G., Shang, Z., & Kuncel, N. (2018). Detecting aberrant behavior and item preknowledge: A comparison of mixture modeling method and residual method. *Journal of Educational and Behavioral Statistics, 43*(4), 469-501.

Wise, S. L. (2017). Rapid-guessing behavior: Its identification, interpretation, and implications. *Educational Measurement: Issues and Practice, 36*(4), 52-61.

Wollack, J. A., & Schoenig, R. W. (2018). Cheating. In B. B. Frey (Ed.), *The Sage encyclopedia of educational research, measurement, and evaluation* (pp. 260–265). Thousand Oaks, CA: Sage.

Zhang, J. (2014). A sequential procedure for detecting compromised items in the item pool of a CAT system. *Applied Psychological Measurement, 38*, 87–104.

Zhang, J., & Li, J. (2016). Monitoring items in real time to enhance CAT security. *Journal of Educational Measurement, 53*, 131–151.

Zhou, T., & Jiao, H. (2022). Exploration of the stacking ensemble machine learning algorithm for cheating detection in large-scale assessment. *Educational and Psychological Measurement*. https://doi.org/10.1177/00131644221117193.

Zhu, X., & Goldberg, A. B. (2009). Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning, 3*(1), 1-130.

Zopluoglu, C. (2019). Detecting examinees with item preknowledge in large-scale testing using extreme gradient boosting (XGBoost). *Educational and Psychological Measurement, 79*(5), 931-961.

Appendix A

Table A1

ART ANOVA Results for False Negative Rate on the Detection of EWP (n=1080)

	SS	*df*	*F*	*p*	Partial η^2
Proportion of EWP (A)	29701013	2	238.06	0.00	0.31
Proportion of CI (B)	57291717	2	718.84	0.00	0.58
Training-Sample Selection Criterion (C)	306569	1	3.62	0.06	0.00
Number of Detections (D)	731433	1	8.68	0.00	0.01
A×B	20875189	4	73.10	0.00	0.22
A×C	1417145	2	8.50	0.00	0.02
B×C	459431	2	2.72	0.07	0.01
A×D	520432	2	3.09	0.05	0.01
B×D	604139	2	3.58	0.03	0.01
C×D	348266	1	4.11	0.04	0.00
A×B×C	2372429	4	7.16	0.00	0.03
A×B×D	468397	4	1.39	0.24	0.01
A×C×D	277212	2	1.64	0.20	0.00
B×C×D	273950	2	1.62	0.20	0.00
A×B×C×D	282439	4	0.83	0.50	0.00

Table A2

ART ANOVA Results for False Positive Rate on the Detection of EWP (n=1080)

	SS	*df*	*F*	*p*	Partial η^2
Proportion of EWP (A)	10366831	2	59.00	0.00	0.10
Proportion of CI (B)	12487722	2	73.90	0.00	0.12
Training-Sample Selection Criterion (C)	6547029	1	73.16	0.00	0.07
Number of Detections (D)	2163126	1	23.35	0.00	0.02
A×B	10069241	4	29.05	0.00	0.10
A×C	3926092	2	21.10	0.00	0.04
B×C	7665280	2	43.12	0.00	0.08
A×D	23479	2	0.12	0.88	0.00
B×D	216317	2	1.14	0.32	0.00
C×D	11434	1	0.12	0.73	0.00
A×B×C	5385499	4	14.64	0.00	0.05
A×B×D	236788	4	0.63	0.64	0.00
A×C×D	16858	2	0.09	0.92	0.00
B×C×D	122825	2	0.65	0.52	0.00
A×B×C×D	79187	4	0.21	0.93	0.00

Table A3

ART ANOVA Results for Precision on the Detection of EWP (n=1080)

	SS	*df*	*F*	*p*	Partial η^2
Proportion of EWP (A)	8967261	2	52.38	0.00	0.09
Proportion of CI (B)	7678163	2	44.97	0.00	0.08
Training-Sample Selection Criterion (C)	2573127	1	28.41	0.00	0.03
Number of Detections (D)	3669402	1	40.85	0.00	0.04
A×B	6854337	4	19.65	0.00	0.07
A×C	1439941	2	7.73	0.00	0.01
B×C	3897153	2	21.63	0.00	0.04
A×D	1848753	2	10.09	0.00	0.02
B×D	145712	2	0.78	0.46	0.00
C×D	116	1	0.00	0.97	0.00
A×B×C	3027639	4	8.26	0.00	0.03
A×B×D	165779	4	0.44	0.78	0.00
A×C×D	7953	2	0.04	0.96	0.00
B×C×D	55772	2	0.30	0.74	0.00
A×B×C×D	50642	4	0.13	0.97	0.00

Table A4

ART ANOVA Results for False Negative Rate on the Detection of CI (n=1080)

	SS	*df*	*F*	*p*	Partial η^2
Proportion of EWP (A)	50509164	2	490.33	0.00	0.48
Proportion of CI (B)	2581513	2	13.47	0.00	0.03
Training-Sample Selection Criterion (C)	106565	1	1.09	0.30	0.00
Number of Detections (D)	1121591	1	11.55	0.00	0.01
A×B	3289297	4	8.65	0.00	0.03
A×C	48205	2	0.25	0.78	0.00
B×C	39797	2	0.20	0.82	0.00
A×D	179629	2	0.92	0.40	0.00
B×D	238805	2	1.22	0.30	0.00
C×D	1040	1	0.01	0.92	0.00
A×B×C	162101	4	0.41	0.80	0.00
A×B×D	254100	4	0.65	0.63	0.00
A×C×D	105601	2	0.54	0.58	0.00
B×C×D	27805	2	0.14	0.87	0.00
A×B×C×D	85427	4	0.22	0.93	0.00

Table A5

ART ANOVA Results for False Positive Rate on the Detection of CI (n=1080)

	SS	*df*	*F*	*p*	Partial η^2
Proportion of EWP (A)	16088999	2	94.97	0.00	0.15
Proportion of CI (B)	31512441	2	228.81	0.00	0.30
Training-Sample Selection Criterion (C)	725200	1	7.34	0.01	0.01
Number of Detections (D)	4763802	1	50.23	0.00	0.05
A×B	11501463	4	32.71	0.00	0.11
A×C	5885970	2	31.54	0.00	0.06
B×C	1147456	2	5.83	0.00	0.01
A×D	68551	2	0.34	0.71	0.00
B×D	1344230	2	6.84	0.00	0.01
C×D	12403	1	0.12	0.72	0.00
A×B×C	2550233	4	6.57	0.00	0.02
A×B×D	70575	4	0.18	0.95	0.00
A×C×D	156760	2	0.79	0.46	0.00
B×C×D	94148	2	0.47	0.62	0.00
A×B×C×D	117050	4	0.29	0.88	0.00

Table A6

ART ANOVA Results for Precision on the Detection of CI (n=1080)

	SS	*df*	*F*	*p*	Partial η^2
Proportion of EWP (A)	28586903	2	197.69	0.00	0.27
Proportion of CI (B)	81089040	2	1794.19	0.00	0.77
Training-Sample Selection Criterion (C)	1420928	1	14.52	0.00	0.01
Number of Detections (D)	6227748	1	66.91	0.00	0.06
A×B	9500767	4	26.35	0.00	0.09
A×C	5657749	2	30.26	0.00	0.05
B×C	1975454	2	10.15	0.00	0.02
A×D	57118	2	0.29	0.75	0.00
B×D	3807213	2	19.92	0.00	0.04
C×D	2023	1	0.02	0.89	0.00
A×B×C	3777997	4	9.85	0.00	0.04
A×B×D	11402	4	0.03	1.00	0.00
A×C×D	192728	2	0.97	0.38	0.00
B×C×D	38983	2	0.20	0.82	0.00
A×B×C×D	154216	4	0.39	0.82	0.00

Appendix B

Figure B1

Standard Deviations for Evaluation Metrics in the Detection of EWP

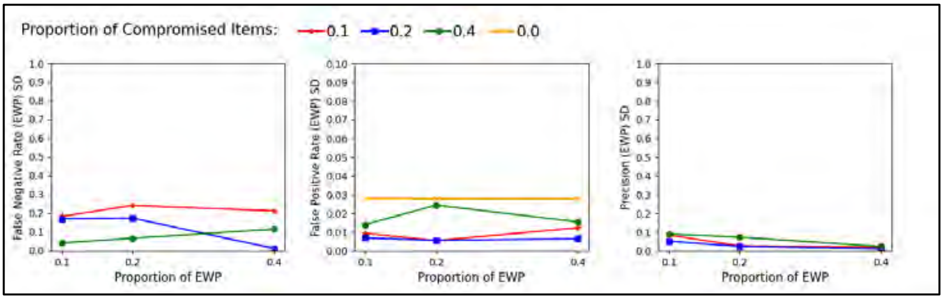
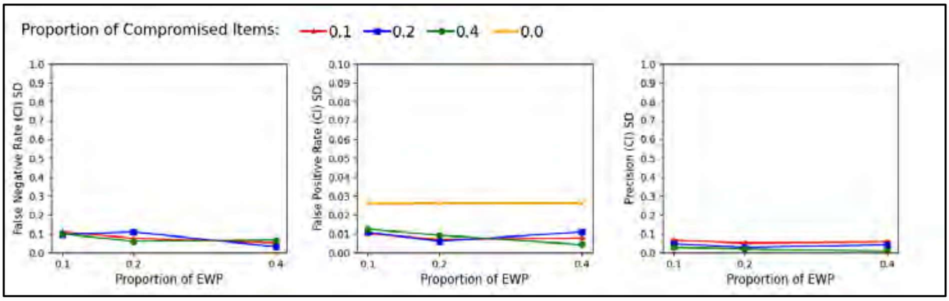


Figure B2

Standard Deviations for Evaluation Metrics in the Detection of CI



Appendix C

Figure C1

Relationship between FNR, FPR and Confidence Score in the Detection of EWP

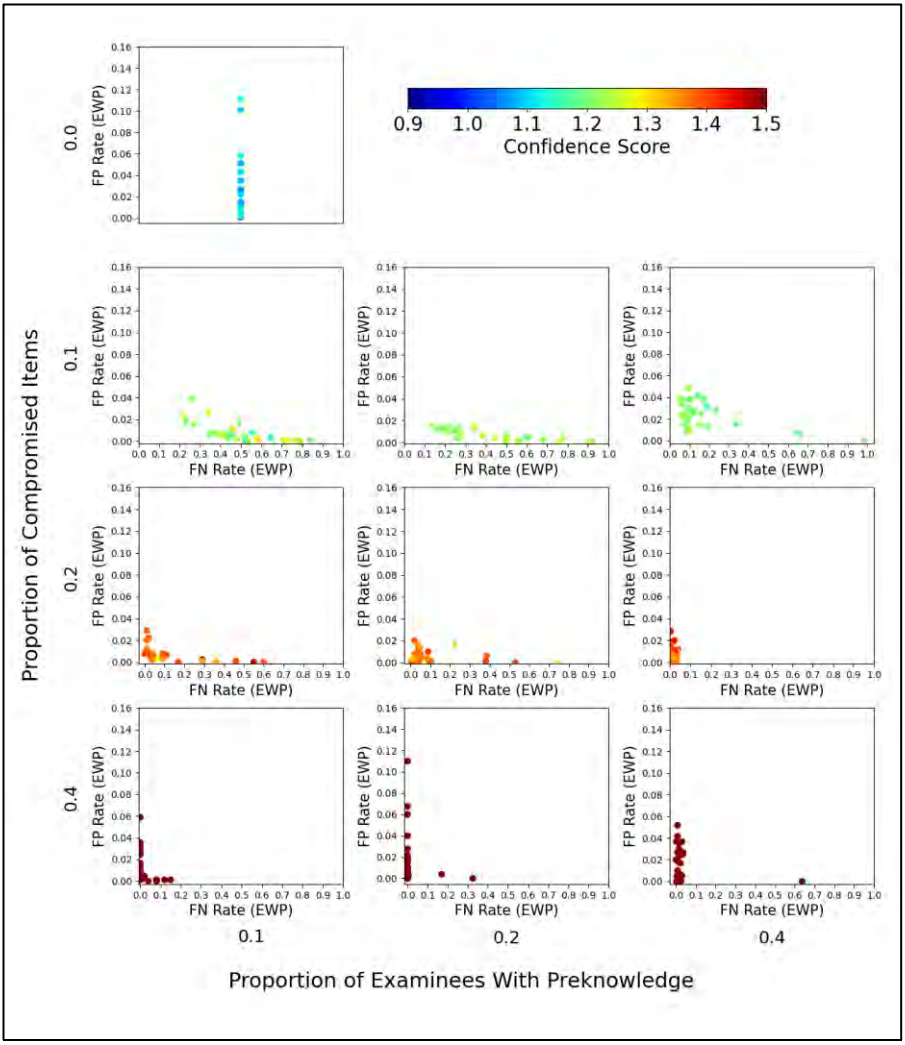
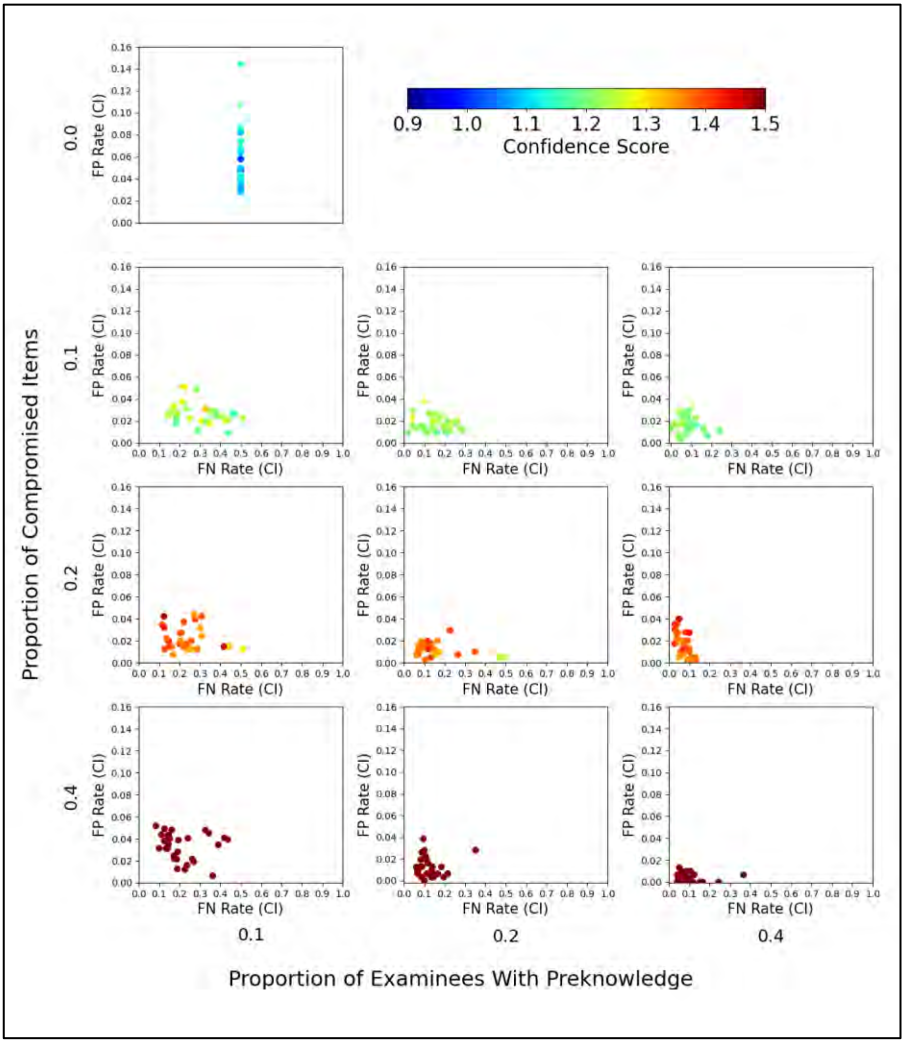


Figure C2

Relationship between FNR, FPR and Confidence Score in the Detection of CI


Confidence Score
Proportion of Compromised Items
FP Rate (CI)
FN Rate (CI)
Proportion of Examinees With Preknowledge