JEM JOURNAL of EDUCATIONAL MEASUREMENT NCME national council on measurement in education

Journal of Educational Measurement
Summer 2019, Vol. 56, No. 2, pp. 251–279

Use of Data Mining Methods to Detect Test Fraud

Kaiwen Man and Jeffrey R. Harring
University of Maryland
Sandip Sinharay
Educational Testing Service

Data mining methods have drawn considerable attention across diverse scientific fields. However, few applications could be found in the areas of psychological and educational measurement, and particularly pertinent to this article, in test security research. In this study, various data mining methods for detecting cheating behaviors on large-scale assessments are explored as an alternative to the traditional methods including person-fit statistics and similarity analysis. A common data set from the Handbook of Quantitative Methods for Detecting Cheating on Tests (Cizek & Wollack) was used for comparing the performance of the different methods. The results indicated that the use of data mining methods may combine multiple sources of information about test takers' performance, which may lead to higher detection rate over traditional item response and response time methods. Several recommendations, all based on our findings, are provided to practitioners.

Cheating or other aberrant test-taking behaviors on psychological and educational tests have been known to undermine the reliability of test usage and the validity of test score interpretations and inferences drawn from these scores (e.g., Cizek & Wollack, 2017; Clark & Desharnais, 1998; Meijer, 1997; van der Linden & Guo, 2008; van Krimpen-Stoop & Meijer, 2001). These undesirable outcomes are exacerbated in high-stakes, competitive assessment scenarios in which fraudulent test-taking behavior not only influences the scoring of the deviant test taker but causes harm to other test takers as these questionable scores impact others' scores with whom they are directly compared (Sinharay, 2017). Momentum is gaining for increased efforts to prevent such behavior from occurring in the first place. More germane to this study, is the call for more effective means of detecting aberrant testing behavior through postassessment forensics. In support of this position is Standard 6.6 of the *Standards for Educational and Psychological Testing* (American Educational Research Association, American Psychological Association, & National Council on Measurement in Education, 2014), which states explicitly that "active efforts to prevent, detect, and correct scores obtained by fraudulent or deceptive means" should be taken.

Several methods have been proposed to examine different types of cheating behavior (see, e.g., Cizek & Wollack, 2017; Kingston & Clark, 2014; Wollack & Fremer, 2013), and typically are classified based on item response theory (IRT) models or response time (RT) models. Proposed methods to detect test fraud generally align with a particular type of cheating behavior. Table 1 summarizes some representative methods for detecting various type of cheating behaviors based on item responses and RTs.

Table 1
Representative Detecting Methods for Various Behaviors

Aberrant Behaviors	Detection Methods
Collusion, Answer Copying, & Preknowledge	*K* (Kling, 1979, cited in Saretsky, 1984); K_1 and K_2 (Sotaridona & Meijer, 2003); *VM* (Belov, 2011); S_2 (Sotaridona & Meijer, 2003); ω (Wollack, 1997); *D* (Trabin & Weiss, 1983); l_z (Drasgow, Levine, & Williams, 1985); Hierarchical RT approach (van der Linden & Guo, 2008); Z_c (Meijer & Sotaridona, 2006); *KL* (Man, Harring, Ouyang, & Thomas, 2018); RT residual analysis (H. Qian, Staniewska, Reckase, & Woo, 2016); Bivariate lognormal RT analysis (van der linden, 2009); l_s (Marianti, Fox, Avetisyan, Veldkamp, & Tijmstra, 2014); l_s^y (Fox & Marianti, 2017); χ_{pt} (Sinharay, 2018)
Suspicious Answer Changing	Linear regression analysis (Primoli, Liassou, Bishop, & Nhouyvanisvong, 2011); Generalized WR analysis (van der Linden & Jeon, 2012); *GBT* (van der Linden & Sotaridona, 2006); *EDI* (Wollack, Cohen, & Eckerly, 2015); *D(G||h)* (Belov, 2015); *PPD_EDI* (Sinharay & Johnson, 2017)
Suspicious Gain Scores	*BHLM* (Skorupski & Egan, 2011); EDI_g(Wollack & Eckerly, 2017)

Methods to detect unexpected gain scores, collusion, preknowledge of items, and other unspecified aberrant test-taking behaviors include the cumulative distribution method (Holland, 2002), l_z^* index (Drasgow, Levine, & McLaughlin, 1987; Snijders, 2001), H^T index (Sijtsma, 1986; Sijtsma & Meijer, 1992), ω (Wollack, 1997), L_s (Sinharay, 2017), erasure detection index (Wollack, Cohen, & Eckerly, 2015), and *S* (Belov, 2015), which are based on individuals' item scores.

RTs have also been used to detect preknowledge of items. For example, van der Linden and Guo (2008) analyzed an RT-based residual that was defined as the discrepancy between the predicted and actual RTs of an examinee's responses. For methods used in other categories of cheating behaviors, interested readers are directed to three recently edited books (Cizek & Wollack, 2017; Kingston & Clark, 2014; Wollack & Fremer, 2013). The use of these methods is limited in some sense because each of these methods can detect only one particular type of cheating behavior. It would be beneficial to find a way to aggregate the detection power of all of these methods across various types of cheating behaviors onto a unified platform. Moreover, many testing programs have transferred from conventional paper-and-pencil tests to computer-based tests and testing environments in the past few years, which has led to the generation of a wealth of test-taker related data (item responses, RTs, and process information) in real time during the administration of tests. It is challenging to use traditional methods to incorporate all the test-taking process information, which involves complex interactions among those variables.

Thus, a question of how best to utilize all of these complementary pieces of information for better pinpointing students who cheat in an efficient and effective manner has not yet been answered.

Due to a big leap in computation capability, data mining methods (e.g., Berkhin, 2006; Hastie, Tibshirani, & Friedman, 2009; Sinharay, 2016a; Strobl, Malley, & Tutz, 2009) have the potential to be effective analytic solutions to address this previously raised question. All variables related to test takers (i.e., item responses, RTs, and biometrical information) could be jointly modeled as input features, thereby capitalizing on all available information rather than just a subset used by any particular traditional test fraud detecting method. Most data mining methods are computationally efficient (and software to implement them are available for free) so that copious amounts of assessment data may be modeled and analyzed in real time.

Many methods currently existing in the field of data mining could be used to assess the various types of processed data generated while students answer questions (e.g., Berkhin, 2006; Chen & Wojcik, 2016; Hastie et al., 2009; P. J. Miller, Lubke, McArtor, & Bergeman, 2016; Strobl et al., 2009). These methods are generally categorized as (1) unsupervised machine learning algorithms and (2) supervised machine learning algorithms. Unsupervised learning methods only rely on the statistical correlation or relative distance measure among the variables used as input to start the algorithm. Variables from different sources including item responses, RTs, and other process data are directly modeled to uncover underlying patterns or hidden structures embedded in the data without precursor model fitting. The data here are known as *unlabeled* data because a classification or categorization is not included (see, e.g., Everitt, 1985; Forgy, 1965; Kohonen, 1982). In contrast, supervised learning methods require the algorithm to learn from a training data set containing input variables and output variables (e.g., classification labels) to produce a prediction function, which could then be used to predict the output variable for given values of the input variables. New data points (test takers in our scenario) would then be classified based on the prediction function (Alpaydin, 2004; Friedberg, 1958; Gillies, 1996).

In this study, various unsupervised and supervised learning methods are explored with a real data set from the *Handbook of Quantitative Methods for Detecting Cheating on Tests* (Cizek & Wollack, 2017) for evaluating the performance of these data mining methods on detecting aberrant test takers who might have preknowledge of certain items or copied answers from their neighbors. The performance of these methods is compared to traditional methods based on item responses and RTs.

Person-Fit Statistics

Proposed methods of aberrant testing behavior detection noted as person-fit statistics (PFSs) were established using IRT models aimed to flag test takers with atypical item responses patterns. The principal idea behind these PFSs is to flag a test takers' response pattern that is not congruent with the response pattern expected from IRT models given the test taker's ability level (Walker, Jennings, & Engelhard, 2018).

PFSs have been used to flag copy-cheating, preknowledge cheating, and careless responding. Through simulation studies, some proposed PFSs such as the nonparametric item response-based H^T statistic (Sijtsma & Meijer, 1992) and *NCI*

statistic (Tatsuoka & Tatsuoka, 1983), the parametric l_z^* statistic (Snijders, 2001), and item RT-based l_t statistic (Marianti et al., 2014) have each shown a high degree of power to detect certain aberrant testing behaviors based on the response patterns of test takers (Dimitrov & Smith, 2006; Karabatsos, 2003; Sijtsma & Meijer, 1992; Sinharay, 2017; Zopluoglu, 2017). Thus, these four representative PFSs are used for comparison with other data mining methods in the upcoming real data analysis.

The two classes of methods—those using IRT modeling of item responses and those using RT data—have been useful in detecting various aberrant testing behaviors with varying degrees of success. As we illustrated earlier, these conventional methods of aberrant test-taking behavior detection have several limitations. Traditional methods such as these are not well-equipped to integrate the vast amount of process data collected as a natural byproduct of computer-based or computer-adaptive testing environments. Data coming from log files as well as other test taker characteristics are continually generated and recorded at regular intervals during an assessment administration and may very well communicate useful and diagnostic evidentiary information to help uncover patterns of aberrant behaviors. Also, each method is used in isolation from every other. Treating aberrant test-taking behavior detection in this manner does not exploit the potential benefit that aggregating such information across methods might reveal.

Data mining algorithms, a class of methods for clustering observations, could be a useful platform for combining information from statistics that can detect different types of aberrant behaviors. Sensitivity to detect aberrant behavior can potentially increase not only by incorporating process and biometrical data as inputs into these algorithms, but also indices based on traditional approaches. Additionally, in contrast to applications involving traditional IRT-based and RT methods, data mining algorithms have the facility to examine both linear and nonlinear relations among variables, thereby increasing flexibility to benefit from modeling interactions between background, psychometric, and biometric data. Several data mining algorithms used in this study are now presented and discussed.

Data Mining Methods

Mining response data to identify clusters of respondents (e.g., such as those who exhibit fraudulent test-taking behavior) is not a new idea in assessment research. Romero, González, Ventura, del Jesús, and Herrera (2009) explained that one must use a data mining strategy that is appropriate for the type of data one wishes to identify, such as data mining to identify patterns of behaviors. Their explanation indicates that data mining can facilitate the identification of cognitive and behavior processes (Berkhin, 2006), and pertinent to the current study, aberrant test-taking behaviors. According to Kerr and Chung (2012), identification of processes within response patterns is typically done with clustering algorithms, which can be classified for the purposes of the current study as either (1) unsupervised machine learning algorithms or (2) supervised machine learning algorithms.

Clustering algorithms—a particular class of unsupervised learning algorithms—are used in this article. Clustering algorithms are processes that use observed similarities or densities in data to identify patterns and group similar observations

(Berkhin, 2006). Three unsupervised learning methods are investigated: (1) K-means clustering, (2) multivariate normal mixture models, and (3) self-organization mapping. We also focus on a category of supervised learning methods whose primary function is accurate classification. The approaches to be investigated are as follows: (1) K-nearest neighbor (KNN), (2) random forests (RFs), and (3) support vector machine (SVM). A description of each method is presented followed by some advantages and disadvantages of each algorithm as they relate to aberrant test-taking behavior detection (see Table 2).

Unsupervised Machine Learning Methods

K-Means Clustering

Although there are several versions of the K-means algorithm, the current research advocates the version defined by Hartigan and Wong (1979), which is generally accepted as the preferred K-means algorithm (Berkhin, 2006; R Core Team, 2014). K-mean clustering attempts to partition *n* observations into K clusters in which each observation belongs to the cluster with the nearest mean. The algorithm begins with a set of K potential centers, which can be defined by the researcher or randomly selected from the data. The choice of initial cluster centers leads to a deterministic partitioning of the space. In other words, K-means will always return the same clustering solution given the same initial cluster centers (e.g., Steinley, 2006). Because the clustering solution relies heavily on where the algorithm launches from, especially for small data sets (Lattin, Carroll, & Green, 2003), some have argued that the algorithm be run multiple times from different starting values to ensure the efficacy of the classification (e.g., Celebi, Kingravi, & Vela, 2013; Khan & Ahmad, 2004).

Once the centers are selected, the algorithm assigns all the test takers to their closest centers and recalculates the new centers defined by these clusters. Distance is determined by a user-specified similarity measure—often the Euclidean distance or Manhattan distance (Fossey, 2017). The algorithm goes through multiple iterations of checking each test taker (e.g., response pattern) to see if it should be moved to a different cluster based on the centers' updated coordinates. If so, it changes the test taker's cluster membership, updates the centers' coordinates, and continues to the next iteration until it converges on a solution where no points are being switched between clusters. The updating process of the K-means approach is illustrated in the panels of Figure 1.

K-means clustering algorithm offers several advantages and disadvantages on aberrant behavior detection. These include the following:

1. K-means algorithm is easy to implement. It only requires practitioners to specify the number of clusters to initiate the algorithm. Usually, in test security investigation, we are expecting to distinguish aberrances from the normally behaved test takers. Thus, two underlying clusters could be reasonably assumed as the number of initial clusters. However, it could also be a disadvantage if users have limited information to determine the number of clusters underlying the data. Thus, K-means method can be used as an exploratory method to manifest more potential subgroups instead of assuming two underlying clusters

Table 2
Advantages and Disadvantages of Various Data Mining Methods for Detecting Aberrant Test Taking Behaviors

	Unsupervised Methods			Supervised Methods		
	K-Means	GFM	SOM	KNN	RF	SVM
PROS	• Easy to implement • Computationally efficient with a high dimensional dataset	• Manifest hidden clusters • Useful to explore subcategories • Show cluster features (volume, shape and orientation)	• Easy to display complex relations of clusters • No distribution assumption	• Effectively robust • Easy to implement	• Facilitate a visual exposition • Runs efficiently on large datasets • Handle higher order variable interactions	• Flexible to apply different kernel functions • Computationally efficient
CONS	• Relies on a predefined distance • Require users to determine number of clusters • Sensitive to the initial cluster centers	• Sensitive to violations of distributional assumptions • Completely exploratory	• Relies on a predefined distance • Require users to define the various parameters (e.g., map size, learning rate)	• Sensitive to redundant and similar features • High computational cost	• Over-fitting • Hard to implement	• Hard to implement • Computationally expensive

Note. GFM = Gaussian finite mixture, SOM = self-organization mapping, KNN = K-nearest neighbor, RF = random forest, SVM = support vector machine.

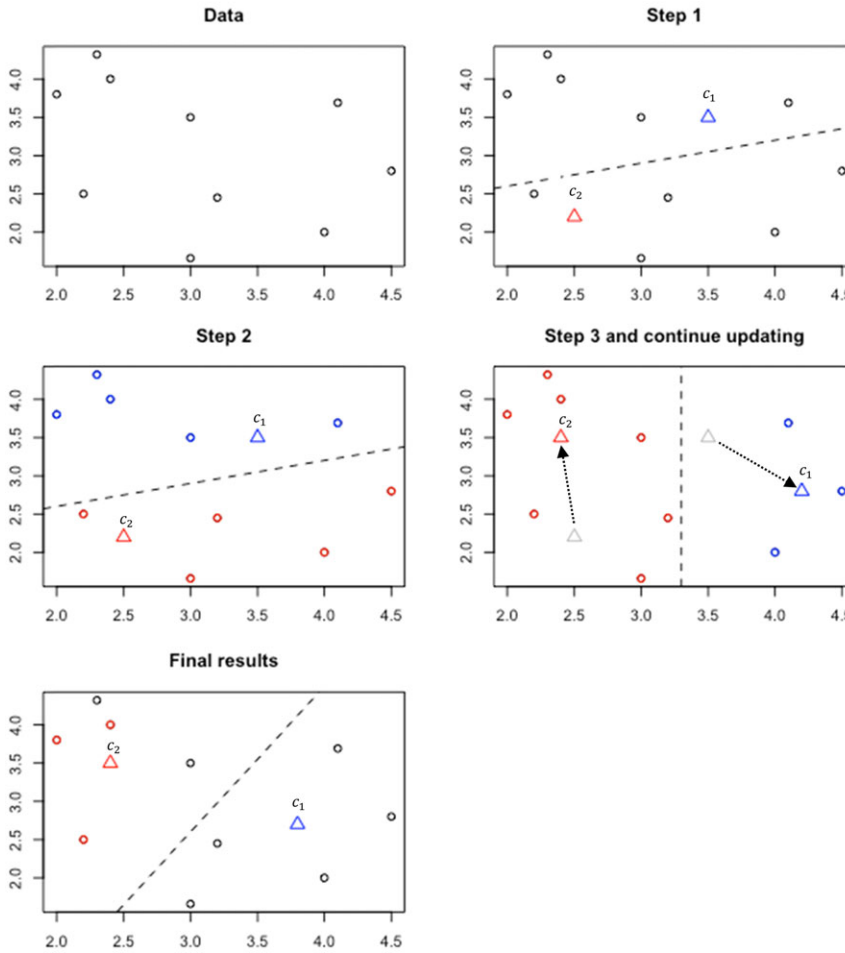

Figure 1. Schematic of iterations of the K-means clustering algorithm: C_1 and C_2 stands for the cluster centers. The dash line indicates the space partition in our example. The dash arrows show the updating process. (Color figure can be viewed at wileyonlinelibrary.com)

(normally behaved or aberrantly behaved test takers) in a data set. By doing so, researchers should summarize the common features associated with the detected subgroups to make further behavioral interpretation.

2. K-means algorithm could be computationally efficient with a high dimensional data set. The algorithm relies on a nonparametric distance measure to classify observations consuming less computational memory than other parametric methods, which requires estimation of model parameters (Hastie et al., 2009). Recently, due to large volumes of process data generated during computer-based testing, K-means could potentially be useful to analyze high-dimensional data for flagging aberrant takers in real time. However, due to the nonparametric nature, the K-means algorithm is sensitive to the initial cluster centers. Many

solutions have been proposed for dealing with this issue (e.g., C. S. Li, 2011; Pena, Lozzano, & Larranaga, 1999). However, these extensions could potentially sacrifice a certain degree of computation efficiency.

Finite Mixture Modeling

A model-based clustering method that might be useful in identifying aberrant test-taking behaviors is finite mixture models, specifically mixtures of multivariate distributions. Mixtures of multivariate distributions (Everitt, 1985; Titterington, Smith, & Makov, 1985) have been applied to a wide range of statistical methodology and take the general form

$$f(\mathbf{s}_j|\varphi, \xi) = \sum_{k=1}^{K} \varphi_k f_k(\mathbf{s}_j|\xi_k), \tag{1}$$

where the composite distribution $\boldsymbol{f}$ on the left-hand side is a mixture of K component densities (e.g., $f_1, \ldots, f_K$) on the right-hand side and $\mathbf{s}_j$ is a p-dimensional vector containing scores for individual j ($j = 1, \ldots, n$) on a set of p observed continuous random variables. Vector $\varphi' = (\varphi_1, \ldots, \varphi_{K-1})$ contains the mixing proportions with the caveat that $0 \leq \varphi_k \leq 1$ for all $k = 1, \ldots, K$ with $\sum_{k=1}^{K} \varphi_k = 1$. Vector $\xi' = (\xi_1', \ldots, \xi_K')$ contains all unknown parameters in all K subpopulations, where $\xi_k' = (\mu_k', vech(\Sigma_k)')$. Operator $vech(\Sigma_k)$ denotes a half-vectorization of a symmetric matrix Σ_k by stacking only the lower triangular part of Σ_k. Following McLachlan and Peel (2000), the kth component density of a mixture of multivariate normal distributions is given by

$$f_k(\mathbf{s}_j|\xi_k) = (2\pi)^{-p/2}|\Sigma_k|^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{s}_j - \mu_k)'\Sigma_k(\mathbf{s}_j - \mu_k)\right\}. \tag{2}$$

One of the main advantages of using a finite mixture modeling (FMM) is that it would manifest hidden clusters embedded in the streams of data by using a likelihood ratio test (e.g., Cox & Hinkley, 1974) or information-based model selection criteria such as Akaike information criterion (AIC) and Bayesian information criterion (BIC; e.g., Anderson & Burnham, 2002). Thus, it would be useful to explore subcategories of aberrant testing behaviors rather than simply focusing on aberrant and normally behaved groups. This could potentially provide more insights for practitioners to understand and investigate specific behavioral groupings. Also, by assuming a multivariate Gaussian density, FMM could reflect the volume, shape, and orientation of each cluster by estimating their corresponding variance–covariance structures. This piece of information could be potentially utilized for understanding characteristics of each identified cluster. For instance, if the fitted Gaussian density contours of each cluster are relatively small and separate, this could indicate strong evidence of the existence of different clusters. Just the opposite would occur if the fitted density contours overlapped too much with each other; the final classification would be doubtful for making the final decision of a number of clusters. Yet, FMM is sensitive to violations of distributional assumptions and is completely exploratory. If the observations do not follow Gaussian distributions, the power of identifying underlying clusters could be decreased.

Self-Organization Mapping

The self-organization mapping (SOM) algorithm (also known as a Kohonen Map) is an artificial neural network algorithm where multidimensional data are mapped to a set of k clusters (or nodes). One of the primary reasons SOM is popular is that the clusters can be mapped to a two-dimensional grid that shows which clusters are similar to each other. This is a valuable tool for visualizing data and validating clusters (Berkhin, 2006).

The SOM algorithm starts with a large learning rate coefficient by randomly selecting a data point, which is used to calculate the distance between the selected data point and any neurons from the user-defined SOM grid. The SOM grid can be either hexagonal or rectangular. After iterating through all the neurons, the neuron that is closest to the selected data point is chosen, which is usually called the best matching unit (BMU). Then, the BMU is pulled closer to the selected data point with a learning rate as well as its neighbor neurons. As time (iterations) progress, the neighborhood around each BMU shrinks so that nearby clusters are not modified when a BMU is updated, and the clusters themselves are not changed as much by looping through a new test taker in the data set because the effect of new test takers is weighted by a decreasing learning algorithm. This is useful in situations where the researcher presents the same cases to the SOM network over and over again to achieve a more stable estimate of cluster centers. As the algorithm runs through its iterations, the learning rate coefficient and the size of the neighborhood shrink until eventually there are only minute, fine-tuning changes to the winning cluster's center (Bullinaria, 2004).

The size of the neighborhood and the rate of decrease can be set by the researcher. The rate of decrease may be linear or nonlinear, and the neighborhood may exist for all of the SOM iterations, or it may be defined so that the neighborhood radius shrinks to a small number after a set number of iterations have been completed. For example, in the default settings of the `som` package (Yan, 2016) in R (R Core Team, 2014) statistical software, the neighborhood's radius is chosen to be larger than two-thirds of the unit-to-unit distances for all of the starting cluster centers. The `som` package then linearly decreases the radius of the neighborhood over one-third of the iterations chosen by the researcher (Wehrens & Buydens, 2007). Once the neighborhood radius diminishes to a small number, clusters near the BMU are no longer updated when cases are reassigned, and the SOM algorithm solution is then identical to the logic used by the K-means algorithm (Kohonen, 1998). The interested readers can visit a website[1] for R tutorial about implementing of `som` functions.

SOM has several benefits for fraudulent testing behavior detection. First, it displays complex high-dimensional topological relations of the cluster centers in a two-dimensional grid, which could be easily visualized and interpreted for test security purposes. Second, SOM does not rely on any assumptions about the distributions of the data, and the solutions are not heavily influenced by outliers (Wehrens & Buydens, 2007). This is because, unlike K-means, SOM never calculates a cluster center's coordinates by taking the mean coordinates of all the test takers assigned to the cluster. Instead, the cluster centers are moved incrementally depending on the case considered at each iteration.

Supervised Machine Learning Methods

K-Nearest Neighbor

KNN is a nonparametric clustering approach representative of supervised learning algorithms and was first proposed by Fix and Hodges (1951). KNN is a straightforward algorithm that attempts to classify new samples (unlabeled observations) by allocating them to the class of the most similar labeled cases by training the machine to learn a function, thereby capturing the relation between the labeled outcome variable and independent variables. The algorithm starts by specifying the size of the neighborhood (K) of a data point by using a distance measure such as Euclidean distance, Manhattan distance, Murkowski distance, or Hamming distance. The choice of K has a significant effect on the KNN results. When K is small, the classification decision would be less stable, and the boundary of separating the different groups would be less linear (James, Witten, Hastie, & Tibshirani, 2013). As K increases, the classification results would be more stable, and the classification boundary would be more linear, which leads to low within-group variance but high classification bias (James et al., 2013). However, this parameter could be tuned to optimize the classification results. Also, K is usually an odd number. Once K is specified, the KNN classifier would identify the K points, which are adjacent to a test observation (a new data point) in the training data set by computing the defined distance between them by looping through the entire data set. The conditional probability for the test observation belonging to a certain class would then be estimated. Finally, the new data point would be allocated to the class with the largest probability. The process would be continued until the last test observation is assigned. Many R-based KNN packages have been created for running the KNN analysis such as KernelKnn (Mouselimis, 2018), *caret* package (Kuhn et al., 2017), and *class* package (Ripley & Venables, 2015). In the current study, the knn function from the *class* package was selected because (1) this package is one of most well-accepted and tested packages for KNN algorithms, and (2) it is also very user-friendly with detailed instructions and documentation that appear in many data mining training websites.[2]

KNN shares many similar advantages as the K-means algorithm, such as simplicity of implementation and flexibility of inputting different type of data. Also, many studies have shown that the KNN method is adequately robust to noisy training data if the training data set is large enough (e.g., Imandoust & Bolandraftar, 2013; Weinberger & Saul, 2009). Therefore, KNN has the potential to generate a stable mapping function, which could be utilized for making accurate and steady classification by limiting the influence of potential outliers. However, it suffers from some limitations. KNN is sensitive to redundant and similar features, which could reduce the classification accuracy (S. Li, Harner, & Adjeroh, 2011; Y. Qian, Yao, & Jia, 2009). In addition, the algorithm has high computational cost if the training data set is large due to calculating the distance of each query to all other inputs in the training data set (Imandoust & Bolandraftar, 2013).

Random Forests

A random forest (RF), a representative ensemble method proposed by Breiman (2001), builds a set of classification and regression trees (CART) to make predictions

by aggregating predicted results from each classification tree. CART, a nonparametric method, recursively segregates the *feature* space (an *n*-dimensional vector space associated with all the predictors) into many small rectangular areas. The CART algorithm splits predictors in a binary manner, meaning each split in the tree-building process only generates two sub-nodes from a parent node. In each sub-node, subjects sharing more homogeneous properties are grouped together. This partitioning process, also called impurity reduction, minimizes the difference between the averaged impurity in the sub-nodes and the impurity in the parent node. Several entropy measures, such as the Gini index, are used to measure the impurity in each sub-node. Each node would be continually split until some stopping conditions are achieved. Commonly used stopping rules of the algorithm include (1) the minimum size of subjects left in a node, (2) a minimum change in the impurity measure after a split, and (3) information criteria such as AIC or BIC (Anderson & Burnham, 2002). After a tree is built, a finalized classification of all the subjects would be predicted in each terminal node. For the RF method, a set of CARTs is built instead of using a single tree to make a prediction. The rationale for this is that a classification prediction based on a single tree would be unstable. For example, if the first splitting variable were chosen differently, the predicted results would be potentially altered especially with a large number of predictors. Moreover, for the RF algorithm, a predictor at each node is randomly selected from the entire feature space for splitting the trees.

In each step of the RF algorithm, either a bootstrap sample or a subset of the entire data set is randomly selected. Thus, by building a diverse set of trees serving as a voting committee would yield more stable and unbiased classification prediction than using a single tree. Voting here means the final prediction is achieved by averaging (weighted or unweighted) the predicted result from each tree. Many other aggregated methods have been developed such as the behavior knowledge space method (Y. S. Huang & Suen, 1995), naive Bayes (NB) combination (Domingos & Pazzani, 1997), and decision templates (Kuncheva, Bezdek, & Duin, 2001). Choice of aggregation method notwithstanding, the ensemble voting method would produce more accurate prediction than using a single tree (e.g., Bauer & Kohavi, 1999; Breiman, 1998). The prediction accuracy could also be checked by an index known as the out-of-bag error rate (Breiman, 1996). Because each tree is built based on either a bootstrapped sample or randomly formed subset of the original data set, the samples are retained so that tree building could be utilized for checking the prediction accuracy. The advantage of using an out-of-bag error rate is that it is a relatively more conservative and precise estimate of the error rate that is closer to the true classification in the population than the overly optimistic result from the prediction by using the original data set (Breiman, 1996; Boulesteix, Strobl, Augustin, & Daumer, 2008). Many R packages have been created for implementing RF algorithms such as Rpart (Therneau & Atkinson, 2018) and tree (Ripley, 2018) and randomForest (Breiman, Cutler, Liaw & Wiener, 2015). In this study, randomForest is used for conducting the analysis.

The RF algorithm has many advantages. Unlike other supervised learning methods, it provides tree-based data representation, which can facilitate a visual understanding of underlying characteristics of classified observations. In test security

investigations, this graphical representation could be further utilized for understanding the behavioral features of aberrantly behaving test takers. Moreover, it also runs efficiently on large data sets and provides the rank of importance of all the features. This piece of information could be helpful to investigate the key factors of classifying aberrant test takers from normally behaved population with increasing efficiency. For instance, by applying the RF algorithm, we could examine momentary responding time to each question, which may indicate suspicious problem-solving behavior—behavior that may reflect a certain degree of preknowledge of the items. Furthermore, the RF algorithm can handle higher order variable interactions reflecting more realistic complex relations among the variables embedded in the data set. Though RF is one of the efficient supervised learning algorithms, some studies have shown that RF can overfit its data set if the stopping rules are not properly set (e.g., Díaz-Uriarte & De Andres, 2006; Segal, 2004).

Support Vector Machine

SVM (Vapnik & Lerner, 1963) has gained popularity as a supervised kernel function–based classification method used in diverse scientific fields (see, e.g., Furey et al., 2000; Hua & Sun, 2001; G. B. Huang, Zhu, & Siew, 2004; Meyer, Leisch, & Hornik, 2003). The SVM algorithm attempts to create an optimal separating boundary, a line, plane, or hyperplane by using a kernel function (linear or nonlinear) that divides the *feature* space (an *n*-dimensional space for predictors) whose margins are maximal. In this regard, this boundary is the best solution out of an infinite possible number of segregating boundaries. The optimal separating boundary, also known as the maximal margin hyperplane, is formed by maximizing the distance between all the training subjects and it. The maximal margin hyperplane is defined by computing the perpendicular distance from each subject to a given separating boundary. The smallest such distance is called the *margin*. As its name suggests, the maximal margin boundary is that separating hyperplane for which the margin is largest. The maximal margin here is also known as the hard margin, which means all the training subjects perfectly lie on either side of the hyperplane without any misclassification. Once the maximal margin hyperplane is constructed based on a training data set, a new test subject could be classified later based on which side of a hyperplane it is located. The hard margin plane, however, is quite sensitive to a change in a subject's data, which may be due to overfitting the training data set. Thus, having a hyperplane that does not perfectly separate all the cases is worthy of attention. Sometimes, this kind of classification hyperplane is also referred to as the soft margin hyperplane, which is more robust to the change of an individual subject. The general support vector classifier can be represented as follows:

$$f(X) = b + \sum_{i \in s} \alpha_i K(x_i, x_{i'}), \tag{3}$$

where $K(x_i, x_{i'})$ is a kernel function that quantifies the similarity of two observations; S is the collection of indices of these support points; α_i and b are parameters needing to be estimated. A simple binary classification example is introduced to help clarify the hard and soft approaches. Suppose a set of n training subjects on p variables exists

$\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^n$, marked with the labels $y_1, \ldots, y_n \in \{-1, 1\}$, is classified into two groups by a linear high-dimensional hyperplane defined as

$$y_i \left(b + \alpha_i \sum_{i=1}^{n} x_{ij,} x_{i'j'} \right) = 0. \tag{4}$$

To find the maximal margin hyperplane, the equation above is optimized based on the constraint to maximize M, subject to

$$\sum_{i=1}^{n} {\alpha_i}^2 = 1, \quad \text{and} \tag{5}$$

$$f(x) = b + \sum_{i \in S} \alpha_i K(x_{i,} x_{i'}) \geq M; \ \text{for all}\, i = 1, \ \ldots, \ n, \tag{6}$$

where M represents the margin of our hyperplane. This is an example of the *hard margin* case requiring each subject in the training set be on the right side of the hyperplane with at least an M margin. The *soft margin* case simply allows the optimization solution to be extended by again maximizing M, subject to

$$\sum_{i=1}^{n} {\alpha_i}^2 = 1$$

$$f(x) = b + \sum_{i \in S} \alpha_i K(x_{i,} x_{i'}) \geq M(1 - \varepsilon_i); \tag{7}$$

$$\varepsilon_i \geq 0 \ \text{ and } \ \sum_{i=1}^{n} \varepsilon_i \leq C, \tag{8}$$

where C is a positive tuning parameter and determines the degree of tolerance of misclassified subjects, which violates the margin. If $C = 0$, the softer margin would transfer to the hard margin case. The term, ε_i $(i = 1, \ldots, n)$, could allow some subjects to be on the incorrect side of the hyperplane. For instance, if $\varepsilon_i > 1$, then the ith observation is on the incorrect side of the hyperplane.

Among many advantages offered by SVM, one of the main benefits of using it is that SVM has the flexibility to select different kernel functions to adequately address practical problems in different modeling scenarios. By applying a proper kernel to the specific scenario, the performance of SVM could be dramatically improved. For example, polynomial or nonlinear kernel functions may be used when the cluster labels and features are nonlinearly related. Many kernels have been created for specific cases such as natural language processing (e.g., string kernels), speech recognition (e.g., time-alignment kernels), and image processing (e.g., histogram intersection kernel). SVM is a flexible platform for identifying aberrances by incorporating more types of data into current detecting framework. For example, writing strings and speech data would be jointly modeled with other psychometric variables like item response and responding time by applying appropriate kernel functions. However, to yield accurate results based on SVM, the tuning parameter and the types of kernel

function should be set properly. In this way, it is relatively harder to be implemented compared with other supervised learning methods.

Identification of Aberrant Test Takers With a Real Data Set

A real data set is analyzed to compare the various detection methods. The data set comes from a credentialing computer-based licensure program, which runs multiple equated forms. Two forms are provided in our data set, Form 1 and Form 2. Each form contains 170 graded items and 10 items served as linking items. Between Form 1 and Form 2, there are 87 common items and 83 scored items that are unique to the form. For the items that are common to the two forms, in all cases, the locations of those items are different. There are 1,636 candidates and 1,644 candidates available for Form 1 and Form 2, respectively. This data set has been cautiously checked because it is known to incorporate test takers who conducted cheating. Some of the candidates were verified as illegally stealing live test content prior to the assessment administration. Other types of misconduct, like cheating during the test, were flagged during the investigation as well. For each form of the assessment, there are approximately 50 candidates (46 for Form 1 and 48 for Form 2) who were flagged as likely cheaters by the testing company. All the flagged test takers are examined through a mix of statistical fraud detection analysis and a serious investigation. Similarly, a number of items were also flagged as having been compromised. Each form contains the following information: (1) item responses, (2) item RTs, (3) test taker identification number, (4) flagged test takers, (5) number of attempts, (6) country of origin, (7) state code, (8) school identification number, (9) center identification number, and (10) total time need to take the assessment.

In this study, three groups of methods introduced previously are used for detecting aberrant test takers: (1) PFSs based on item responses and RTs, (2) unsupervised learning methods, and (3) supervised learning methods. The item response and RT-based methods are currently utilized in the industry as approaches to identify test takers showing fraudulent behavior, while unsupervised and supervised learning methods have yet to be fully deployed and investigated.

IRT-Based and RT Methods

PFSs in IRT have been created to examine the discrepancies between item responses for an individual compared to the average response profile and many refinements have been made over the years (see, e.g., Meijer & Sijtsma, 2001; Sinharay, 2016b). Cutoff values were used to differentiate "atypical" response profiles (i.e., aberrant responses) different from what one would expect given their ability level computed under a given IRT model. The cutoffs for different PFSs were determined by their underlying null empirical distribution, which could be either mathematically derived or approximated by running Monte Carlo simulations. For example, cutoff values for the l_z^* statistic (Snijders, 2001) follow a standard normal distribution that was mathematically derived. With a particular level of significance, α, a one-sided critical value could be determined from the standard normal distribution. For example, if $\alpha = 0.05$, a cutoff value would be set at –1.645 in which any case exhibiting a more extreme value of l_z^*would necessarily be flagged as aberrant. Similarly, Sijtsma

and Meijer (1992) suggested .3 as the recommended cutoff for the H^T statistic, although no universal standard exists. This may be due to the fact that both test length and underlying ability distribution of the test takers can impact the null distribution of the statistic. For example, Karabatsos (2003) suggested .22 as the cutoff for the H^T statistic. For the *NCI* index, T. W. Huang (2012) suggested a cutoff of approximately 0 for flagging aberrant test takers.

Cutoffs in the current study. For the traditional IRT-based and RT-based approaches, several methodological studies have suggested particular cutoff values and thresholds for identifying aberrant test-taking behaviors. However, these benchmarks depend on both test length (i.e., number of items) as well as the underlying ability distribution of the examinees. There appears to be no universally agreed-upon standard to use for every testing scenario. A strategy employed here was to use the test length of the real data example—description forthcoming—and the underlying ability distribution of the 1,000+ examinees from this data set, to create empirical sampling null distributions for each index with these assessment characteristics. To make the cutoffs comparable across indices, we chose cutoff values corresponding to partitioning the lower 5% of test takers (i.e., $\alpha = 0.05$), which were then identified as aberrant. The method of how the sampling distributions were created is outlined as follows:

1. We used the *cutoff* function in the R package PerFit (Tendeiro, Meijer, & Niessen, 2016) to simulate 50,000 test takers. The *cutoff* function takes in (0, 1) response data from *J* test takers across *I* items as input and fits a Rasch model to it resulting in estimated difficulty item parameters for each item and predicted ability parameters for each test taker. For the real data example, this meant response data on $I = 170$ items across $J = 1636$ test takers.
2. The *cutoff* function then simulates dichotomous response data according to the Rasch model for a single hypothetical test taker using the estimated item parameters by randomly drawing a single ability parameter from the original pool of 1,636 predicted ability parameters in Step 1. With replacement, the algorithm continues in this fashion creating response data for 170 items for 50,000 hypothetical test takers.
3. The *cutoff* function computes the index of interest—one for each hypothetical test taker—and returns the value of the empirically generated distribution that cuts off the bottom 5%.

Using this simulation strategy, thresholds used in this study were –1.665 (95% confidence interval [CI] [–1.6826, –1.6431]), .108 (95% CI [.1073, .1087]), and .28 (95% CI [.2790, .2835]) for the l_z^*, H^T, and NCI indices on form A, respectively. For form B, the cutoffs for the same indices were –1.709 (95% CI [–1.7932, –1.6377]), .105 (95% CI [.0995, .1109]), and .281 (95% CI [.2784, .2811]), respectively. The figures of these empirical null distributions for the calculated indices are available on our website.[3]

For the RT-based PFS, the l_z^* statistic follows a chi-square distribution with degrees of freedom equal to the number of items. Utilizing a similar procedure as outlined above but using self-generated code to fit the RT model, a threshold of 202 was

determined to cut off the top 5% of test takers. The empirical null distribution is listed on our website.[4]

Data Mining Methods

For the unsupervised learning methods, all test takers were classified when the unsupervised learning algorithm reached convergence. Different unsupervised learning methods have various algorithm-stopping rules. The K-means method uses a measure called within-cluster variation (WCV) as its convergence measure. Once the WCV is minimized to a certain level, the classification of all the subjects would be finalized. Figure 2 shows two classified groups, one group of test takers flagged as aberrant marked by the black dots and another normally behaved group of test takers marked with dark gray-colored dots. The black dots are inferred as aberrant test takers because most of them have unreasonably short total RTs, which are less than 39 minutes. Nevertheless, the time limit for this credentialing test was 2 hours. Second, based on the cutoffs of the PFSs, which are a subset of the overall number of input variables for the K-means algorithm, the majority of the black-dotted test takers are flagged as aberrances based on those PFSs. Finally, in this data set, the majority of black-dotted test takers are marked as actually fraudulent test takers identified a priori by the testing company.

The finite multivariate mixture model uses a maximum likelihood estimation method to estimate the proportion parameter for a two-class mixture, which are labeled as aberrant and nonaberrant test taking groups. At convergence of the algorithm, the test takers are classified into the class that corresponds to their highest posterior probabilities (Harring & Hodis, 2016). The SOM algorithm uses the maximum iterations as the default-stopping rule. Typically, the number of iterations is substantially large so that the classification based on SOM is stable. Once the SOM algorithm achieves convergence, a classification decision is made for all the subjects. Figure 3 shows two groups classified based on SOM. The test takers colored in light gray are those labeled as aberrant test takers, while the test takers in dark gray represent normal test takers.

For the supervised learning methods, the predicted response classes are based on well-trained models. Two-thirds of the subjects in the sample with their true class labels were initially used for training a specific model. The remaining one-third of the data set was used for prediction and model validation. In general, a model would be fitted first. Then, a new test subject would be classified based on the fitted model. Data normalization, also called feature scaling, is a process to transfer the range of different independent variables or features onto a common scale are performed for the supervised learning methods in this study (see Vapnik, 2005, for a discussion of the advantages for supervised learning methods). For traditional and unsupervised learning methods, however, data normalization was not performed because these data are either invariant to monotonic transformations of individual features or not recommended by some studies. Also, many traditional methods are parameter model-based clustering methods, which do not rely on the geometric distance measures for classification. Thus, it is not necessary to implement feature normalization (see, e.g., Dubes & Jain, 1988; Hastie et al., 2009; Strobl et al., 2009).

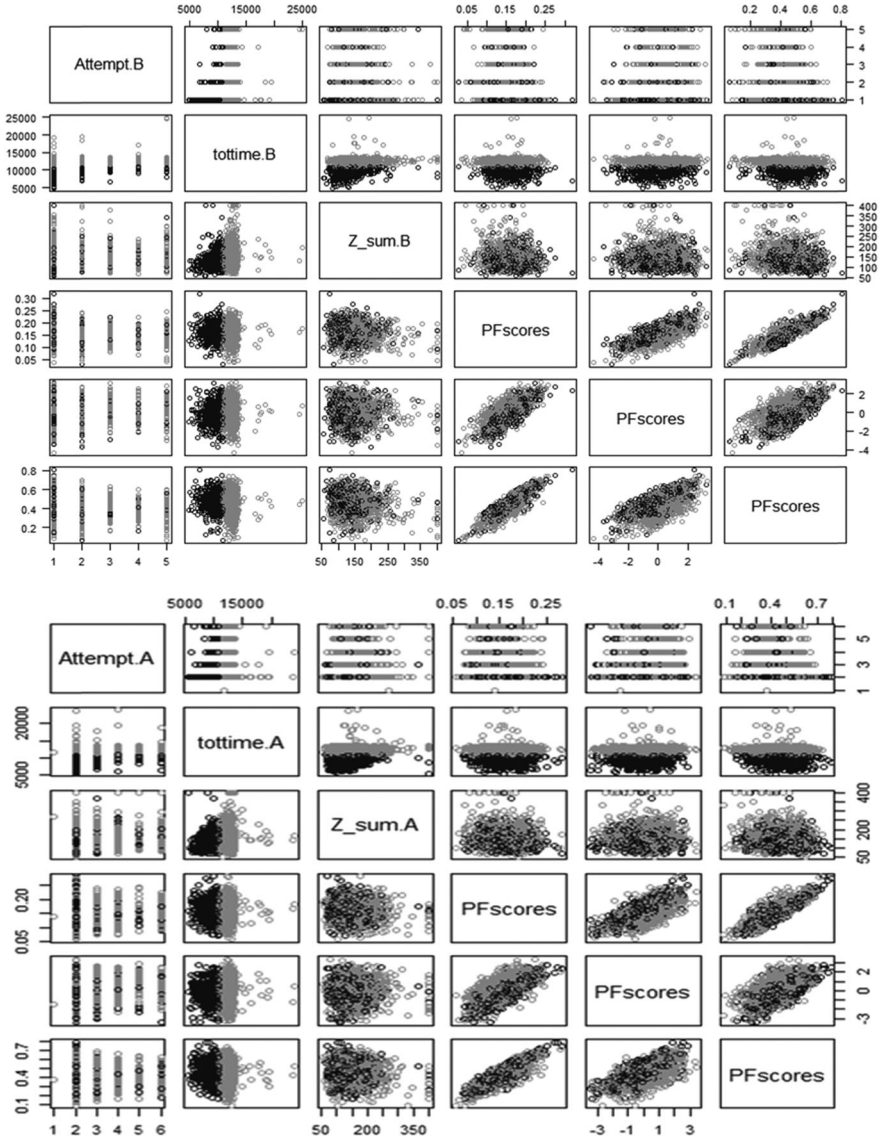

Figure 2. Scatterplot of key variables for Form 1 based on K-means method and the Scatterplot of key variables for Form 2 based on K-means method.
Note. Attempt A and Attempt B indicate the number of attempts a test taker took his/her exam. Tottime A and Tottime B indicate the total time used by a test taker for finishing his/her exam. Z_sumA and Z_sumB stands for the RT-based indexes *lt*. PSFscore indicate the item response based index: Ht, l^*z, and NCI.

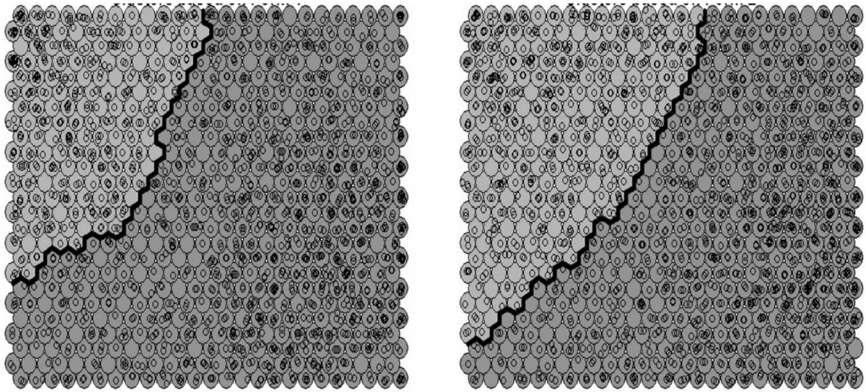

Figure 3. SOM classification based on two forms.

Four types of scaling methods have drawn much attention in the practical usage, which are scaling by variance, mean normalization, scaling by minimum and maximum values, and scaling to unit length (see, e.g., Fukunaga, 2013; Juszczak, Tax, & Duin, 2002). In this study, we scaled the data by minimum and maximum values implying that all independent variables are scaled to the range [0, 1] by computing $(x - \min(x))/[\max(x) - \min(x)]$. An advantage of this type of scaling is that it can accommodate binary item responses.

The KNN method classifies new test takers based on the class label of their closest neighbors in the training data set. Based on this algorithm, each new subject could be assigned to a given class. In our example, a test subject would be classified into either an aberrantly behaved group or a normally behaved group. The RF algorithm predicts a response class in each terminal node of the tree by counting for the most frequent response class in the classification trees. The predicted aberrant test takers, in our examples, are those labeled with a 1 instead of 0. For instance, Figure 4 shows that those who spent time less than 1.6 seconds on the item 58 were more likely to be classified as aberrant, which are the test takers marked with 1 on the left corner node. Other terminal nodes could be interpreted similarly. SVM attempts to generate a classification boundary by fitting a particular kernel function. Once the kernel function is fitted, a test subject could be classified based on which side of the classification boundary it is located. The boundary in our example separates aberrant from normal test takers.

Feature Selection

After data normalization to make inferences from the model, a set of features $\{x_{\{1\}}, x_{\{2\}}, \ldots, x_{\{m\}}\}$, also called independent variables, and attributes are selected from the total number of potential input features $\{x_{\{1\}}, x_{\{2\}}, \ldots, x_{\{M\}}\}$, where $m < M$. It is essentially a filtering process, which has been discussed in the area of data mining for some time (John, Kohavi, & Pfleger, 1994; Koller & Sahami, 1996; A. J. Miller, 1990). Implementing this selection process increases the interpretability of the model, which is frequently less complex and more parsimonious. In this study,

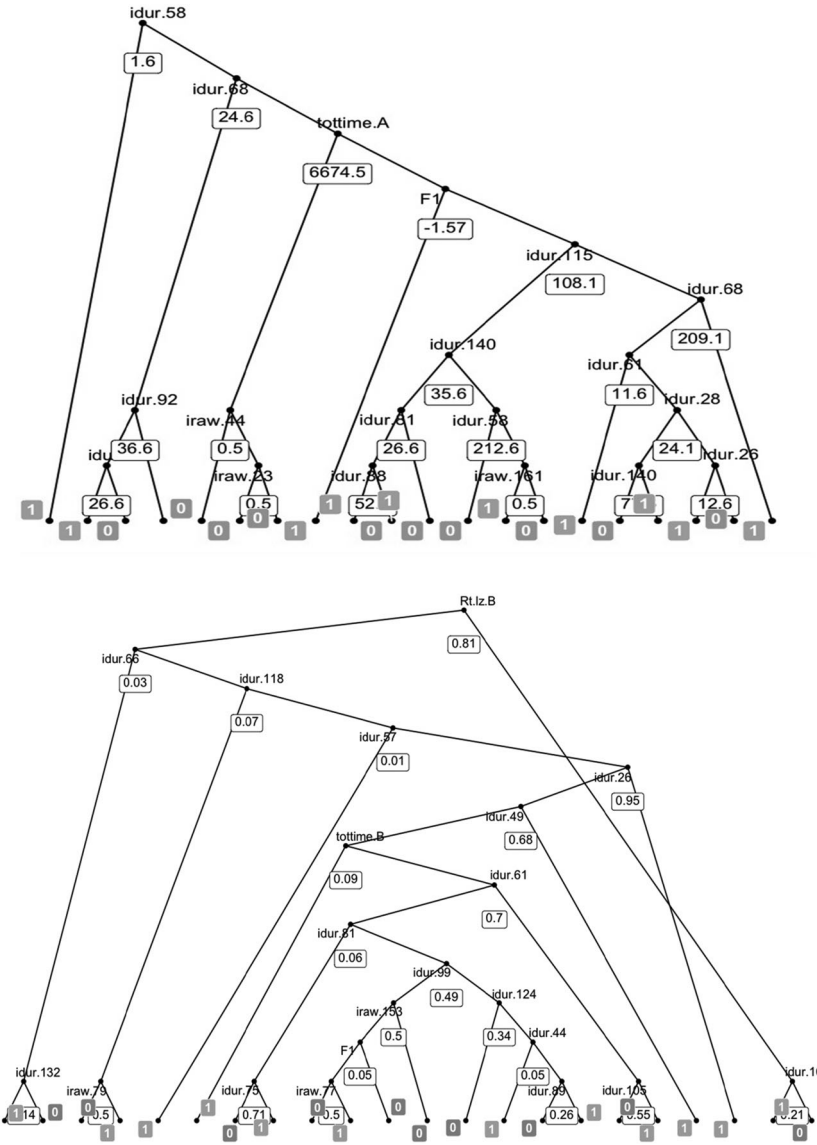

Figure 4. Binary classification trees: Partitions of the Form 1 data and Form 2 data.

two "filtering methods" are used as feature selection methods: (1) Pearson correlation between any pair of input variables and (2) the variable importance index (VII). VII works by randomly permuting the values of a feature (input) variable, which breaks the original relation between the variable and other variables. Then, the permuted feature is used again with other unpermuted features for making predictions. If the prediction accuracy decreases, the gap before and after the permutation on a specific variable, averaged across all the trees, would be used as a measure of variable importance.

Table 3
Sensitivity and Specificity for PFS IRT- and RT-Based Methods for Parallel Forms

% Consistent Decision	Sensitivity	Specificity	Sensitivity	Specificity
	Form 1		Form 2	
H^T PFS	7%	90%	21%	91%
l_z^* PFS	9%	90%	17%	91%
NCI PFS	13%	92%	17%	92%
l^t PFS	35%	83%	25%	87%

Based on the methods mentioned above, the final set of variables that are used in the analyses includes (1) item responses, (2) item RTs, (3) number of attempts, (4) total time, (5) H^T PSF index, (6) l_z^* PSF index, (7) *NCI* PSF index, (8) l_tRT index, and (9) latent speediness (τ_j) for each test taker. Although the correlation between some features was relatively high (i.e., $r = .87$ between the H^T and l_z^*), the VII showed all variables were important to include yielding high classification accuracy.[5]

Results

The testing company providing the data had identified aberrant (and nonaberrant) test takers, and their ID numbers were known. For each of the proposed methods, a method-based classification of aberrant and nonaberrant test takers was obtained. Sensitivity and specificity are used as outcome measures for evaluating the performance of different methods. Sensitivity is defined here as the percentage of test takers that were identified as aberrant by the testing company and that were classified as aberrant by the particular method. It would be calculated as $100\% \times [TP/(TP + FP)]$. TP stands for the true positive (i.e., TP are those test takers correctly classified as aberrant) and FP stands for the false positive (i.e., FP are those test takers incorrectly classified as nonaberrant). Specificity, on the other hand, is defined here as percentage of test takers that were identified as nonaberrant ("normal") and that were classified as nonaberrant by the particular method. Again, specificity would be computed as $100\% \times [TP/(TN + FN)]$. TN stands for the true negative, and FN stands for the false negative.

As a means of comparison of the performance of aberrant test-taking behavior detection, Table 3 shows the sensitivity and specificity rates for various traditional item response and RT-based methods across two parallel forms of the assessment. The proportions of test takers were statistically significant at the 5% level. The four rows of numbers provided the proportions of successfully detected aberrant test takers and the proportions of safely "guarded" normally behaved test takers. For example, the H^T PFS successfully flagged 7% and 21% of aberrant test takers who were flagged by the licensure testing company. Table 3 also suggests that the item response–based methods (first three rows) had relatively low sensitivity rates compared to the item RT-based l_t PFS, which indicates that the RT index was better at detecting aberrant test takers than the IRT-based indices. However, the item RT-based l_t PFS had a lower specificity rate compared with other IRT-based indices, which suggests that

Table 4
Sensitivity and Specificity for Unsupervised Learning Methods for Parallel Forms

% Consistent Decision	Sensitivity	Specificity	Sensitivity	Specificity
	Form 1		Form 2	
K-Means	46%	69%	54%	69%
Gaussian Finite Mixture	43%	70%	50%	70%
Self-Organization Mapping	46%	73%	52%	73%

Table 5
Sensitivity and Specificity for Supervised Machine Learning Methods for Parallel Forms

% Consistent Decision	Sensitivity	Specificity	Sensitivity	Specificity
	Form 1		Form 2	
K-Nearest Neighbor	44%	97%	41%	98%
Random Forest	46%	99%	47%	99%
Supported Vector Machine	46%	99%	35%	99%

these latter indices would likely flag a larger proportion of normally behaved test takers as cheaters than using an index based on RTs.

Table 4 presents the comparison of the capacities on detecting aberrantly behaved test takers across different unsupervised learning methods based on the two parallel licensure tests. The overall sensitivity rates averaged across the two forms based on the unsupervised learning methods were relatively higher than the sensitivity rates based on the traditional IRT- and RT-based methods. The highest sensitivity rate was approximately 54%, while the lowest rate was 43%. Among all the proposed unsupervised methods, the K-means method generated relatively higher sensitivity rates than other unsupervised methods, which was about 50% across the two forms. However, more normally behaved test takers were flagged as cheaters based on the unsupervised learning methods by comparing the specificity rates with those from the traditional methods displayed in Table 4. Sensitivity and specificity rates for supervised learning methods are presented in Table 4.

For the supervised learning methods, the results were summarized based on 10-folds cross-validation. Table 5 indicates that the overall sensitivity rates based on the supervised learning methods were similar to the ones based on the unsupervised learning methods. The specificity, however, was much higher than the proposed traditional and unsupervised methods, which suggests that the supervised learning methods could potentially protect normally behaved test takers from incorrect identification with the promising power of capturing real cheaters.

Discussion

Identification of fraudulent test takers, especially for high-stakes assessments, is a predominant, contemporary issue that many testing companies are currently grappling with. The inclusion of scores from aberrant-behaved test takers has

been known to compromise the consistency of test usage, the validity of test score interpretations, and inferences drawn from these scores. Current practice relies on the implementation of postassessment psychometric forensic analyses via traditional detection methods utilizing test takers item responses, and, when process data are collected, their RTs as well. Both parametric and nonparametric IRT-based PFSs as well as an RT-based index that have had a historical presence in the methodological literature, and have been shown to be somewhat beneficial in identifying particular types of fraudulent behaviors, were reviewed. Existing approaches to detect test fraud involve the use of these statistics in isolation—not exploiting the advantage of combining their collective power—because certain indices only detect one type of aberrant behavior. Data mining methods in the form of unsupervised and supervised machine learning algorithms allow investigators to combine information from several sources/approaches to more accurately detect test fraud.

These newer methods are well-suited to tackle identification of fraudulent test-taking behaviors as they can incorporate vast amounts of data coming from potentially numerous different, rich sources (e.g., process data, biometrical data, and psychometric data). This point is particularly salient as many testing administrations are moving away from pencil-and-paper assessments toward computer-based environments.

This study used response data from a certification licensure exam in which *cheaters* had been already identified by the testing organization. This provided the ability to evaluate the capabilities of numerous algorithms and methods in classifying these same test takers. The findings from the real data analyses showed that the data mining methods investigated here gave relatively high detection rates (sensitivity) than traditional methods—especially the supervised methods—which were able to detect the aberrant test takers without incorrectly flagging normally behaved test takers as cheaters. Potentially, by increasing the size of the pool of identified cheaters, the detection accuracy based on supervised learning methods could also be improved. This is because the machine (algorithm) could better learn the patterns from greater numbers of test takers thereby refining its ability to detect the aberrant test takers. However, to take advantage of supervised methods on cheating behavior detection, testing companies have to create a "blacklist," which are those test takers who have been flagged from a serious investigation of previous assessments. This can likely be implemented for assessments that have sustained longevity in the practice (e.g., GRE). On the other hand, for newly developed assessments lacking repeated, continual administrations, traditional and unsupervised learning methods could very well play an important role for building a preliminary pool of suspicious test taker behavior. This type of initial screening and flagging of aberrant test-taking behaviors could be used to train the supervised learning methods, which would undoubtedly become more knowledgeable with updating occurring with more test administrations. Yet, in order to properly determine the correct label of classified groups, such as "cheaters" and "noncheaters," based on the unsupervised learning methods, practitioners need to examine the inferred cases closely either based on traditional detection methods or other practical evidence collected or reported during the test.

The findings from analyses performed on the real data suggest that data mining methods warrant further methodological investigation to ascertain under what

realistic testing conditions they are best suited. Positively, data mining methods are able to accommodate complex data sets with large sample sizes (n) and large numbers of variables (p), but can unexpectedly be useful when n is small compared to p. In this latter scenario, parametric methods may be useless or may yield unstable parameter estimates due to the limited sample size. Many variables, such as biometric variables—data coming from eye-tracking analyses—or background variables can be integrated into the modeling framework. Indeed, a multitude of data can be analyzed with high efficiency because most data mining methods, including the ones used in this study, are nonparametric distance-based methods, which require few parameters to be estimated.

To flag aberrant test-taking behaviors with increased accuracy, multiple sources of information about test takers including data stemming from bioinformation technologies could be integrated based on the data mining methods. Ideally, all of this information could be the inputs to be aggregated using highly efficient computational methods such as cloud computing. Mislevy (2016) indicated that new forms of assessments would involve psychometric models, bioinformation, machine learning, and data mining methods using a high-efficiency computation platform. Through methodological investigations and analyses using empirical data, the signal-to-noise ratio (SNR) could be increased—meaning that greater classification accuracy could be achieved with more sensitivity to nuanced behaviors.

Clearly, the reality is much more complex than any model could adequately capture. A high-efficiency modeling framework like those for many data mining methods could be used to uncover subtle, hidden patterns in streams of data—letting the data speak for themselves. This may very well be the most effective way for detecting aberrant test taker behaviors, which on their face, can be quite difficult to distinguish from normal testing behaviors.

Notes

[1]https://clarkdatalabs.github.io/soms/SOM_NBA

[2]One such website is http://vision.stanford.edu/teaching/cs231n-demos/knn/.

[3]https://www.researchgate.net/profile/Kaiwen_Man2

[4]https://www.researchgate.net/profile/Kaiwen_Man2

[5]Variable importance figure is available in our website: https://www.researchgate.net/profile/Kaiwen_Man2

References

Alpaydin, E. (2004). *Introduction to machine learning*. Cambridge, MA: MIT Press.

American Educational Research Association, American Psychological Association, & National Council on Measurement in Education. (2014). *Standards for educational and psychological testing*. Washington, DC: American Educational Research Association.

Anderson, D., & Burnham, K. (2002). Avoiding pitfalls when using information-theoretic methods. *Journal of Wildlife Management*, *66*, 912–918.

Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, *36*, 105–139.

Belov, D. I. (2011). Detection of answer copying based on the structure of a high-stakes test. *Applied Psychological Measurement*, *35*, 495–517.

Belov, D. I. (2013). Detection of test collusion via Kullback–Leibler divergence. *Journal of Educational Measurement*, *50*, 141–163.

Belov, D. I. (2015). Robust detection of examinees with aberrant answer changes. *Journal of Educational Measurement*, *52*, 437–456.

Berkhin, P. (2006). A survey of clustering data mining techniques. In J. Kogan, C. Nicholas, & M. Teboulle (Eds.), *Grouping multidimensional data* (pp. 25–71). Berlin, Germany: Springer.

Boulesteix, A. L., Strobl, C., Augustin, T., & Daumer, M. (2008). Evaluating microarray-based classifiers: An overview. *Cancer Informatics*, *6*, 77–97.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*, 123–140.

Breiman, L. (1998). Arcing classifier (with discussion and a rejoinder by the author). *Annals of Statistics*, *26*, 801–849.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32.

Breiman, L., Cutler, A., Liaw, A., & Wiener, M. (2015). randomForest: Breiman and Cutler's Random Forests for Classification and Regression. R package version 4.6-12. Retrieved from https://CRAN.R-project.org/package=randomForest

Bullinaria, J. A. (2004). Introduction to neural networks. In M. N. José & M. F. Santos (Eds.), *School of computer science* (pp. 512–523). Birmingham, UK: Springer.

Celebi, M. E., Kingravi, H. A., & Vela, P. A. (2013). A comparative study of efficient initialization methods for the K-means clustering algorithm. *Expert Systems With Applications*, *40*(1), 200–210.

Chen, E. E., & Wojcik, S. P. (2016). A practical guide to big data research in psychology. *Psychological Methods*, *21*, 458–474.

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

Clark, S. J., & Desharnais, R. A. (1998). Honest answers to embarrassing questions: Detecting cheating in the randomized response model. *Psychological Methods*, *3*(2), 160–168.

Cox, D. R., & Hinkley, D. V. (1974). *Theoretical statistics*. London, UK: Chapman & Hall.

Díaz-Uriarte, R., & De Andres, S. A. (2006). Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, *7*(1), 3. https://doi.org/10.1186/1471-2105-7-3

Dimitrov, D. M., & Smith, R. M. (2006). Adjusted Rasch person-fit statistics. *Journal of Applied Measurement*, *7*, 170–183.

Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, *29*, 103–130.

Drasgow, F., Levine, M. V., & McLaughlin, M. E. (1987). Detecting inappropriate test scores with optimal and practical appropriateness indices. *Applied Psychological Measurement*, *11*, 59–79.

Drasgow, F., Levine, M. V., & 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.

Dubes, R. C., & Jain, A. K. (1988). *Algorithms for clustering data*. Upper Saddle River, NJ: Prentice-Hall.

Everitt, B. S. (1985). Mixture distributions. In S. Kotz & N. L. Johnson (Eds.), *Encyclopedia of statistical sciences* (pp. 559–569). New York, NY: John Wiley.

Fix, E., & Hodges, Jr., J. L. (1951). Discriminatory analysis-nonparametric discrimination: Consistency properties. *International Statistical Review*, *3*, 238–247.

Forgy, E. W. (1965). Cluster analysis of multivariate data: Efficiency versus interpretability of classifications. *Biometrics*, *21*, 768–769.

Fossey, W. A. (2017). *An evaluation of clustering algorithms for modeling game-based assessment work processes* (Unpublished doctoral dissertation). University of Maryland, College Park, MD.

Fox, J. P., & Marianti, S. (2017). Person-fit statistics for joint models for accuracy and speed. *Journal of Educational Measurement*, *54*, 243–262.

Friedberg, R. M. (1958). A learning machine: Part I. *Journal of Research and Development*, *2*, 2–13.

Fukunaga, K. (2013). *Introduction to statistical pattern recognition*. San Diego, CA: Academic Press.

Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., & Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, *16*, 906–914.

Gillies, D. (1996). *Artificial intelligence and scientific method*. Oxford, UK: Oxford University Press.

Harring, J. R., & Hodis, F. A. (2016). Mixture modeling: Applications in educational psychology. *Educational Psychologist*, *51*, 354–367.

Hartigan, J. A., & Wong, M. A. (1979). A K-means clustering algorithm. *Applied Statistics*, *28*, 100–108.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning*. New York, NY: Springer.

Holland, P. W. (2002). Two measures of change in the gaps between the CDFs of test-score distributions. *Journal of Educational and Behavioral Statistics*, *27*, 3–17.

Hua, S., & Sun, Z. (2001). Support vector machine approach for protein subcellular localization prediction. *Bioinformatics*, *17*, 721–728.

Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2004). Extreme learning machine: A new learning scheme of feedforward neural networks. *Proceedings of IEEE International Joint Conference on Neural Networks* (pp. 163–171). Mahwah, NJ: Lawrence Erlbaum.

Huang, T. W. (2012). Aberrance detection powers of the BW and person-fit indices. *Educational Technology and Society*, *15*, 28–37.

Huang, Y. S., & Suen, C. Y. (1995). A method of combining multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *17*, 90–94.

Imandoust, S. B., & Bolandraftar, M. (2013). Application of K-nearest neighbor (KNN) approach for predicting economic events: Theoretical background. *International Journal of Engineering Research and Applications*, *3*, 605–610.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning*. New York, NY: Springer.

Juszczak, P., Tax, D., Duin, R. P. W. (2002). Feature scaling in support vector data description. In E. Deprettere, A. Belloum, J. Heijnsdijk, & F. van der Stappen (Eds.), *Proceedings of the ASCI 2002 8th Annual Conference of the advanced school for computing and imaging* (pp. 95–102). Lochem, The Netherlands: Advanced School for Computing and Imaging.

John, G. H., Kohavi, R., & Pfleger, K. (1994). Irrelevant features and the subset selection problem. In W.W. Cohen (Ed.), *Proceedings of Eleventh International Conference* (pp. 121–129). New Brunswick, NJ: Elsevier.

Karabatsos, G. (2003). Comparing the aberrant response detection performance of thirty-six person-fit statistics. *Applied Measurement in Education*, *16*, 277–298.

Kerr, D., & Chung, G. K. (2012). Identifying key features of student performance in educational video games and simulations through cluster analysis. *Journal of Educational Data Mining*, *4*, 144–182.

Khan, S. S., & Ahmad, A. (2004). Cluster center initialization algorithm for K-means clustering. *Pattern Recognition Letters*, *25*, 1293–1302.

Kingston, N., & Clark, A. (Eds.). (2014). *Test fraud: Statistical detection and methodology*. London, UK: Routledge.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, *43*, 59–69.

Kohonen, T. (1998). The self-organizing map. *Neurocomputing*, *21*, 1–6.

Koller, D., & Sahami, M. (1996). Toward optimal feature selection. *Stanford InfoLab*, *77*, 1–15.

Kuhn, M. Contributions from Wing, J., Weston, S., Williams, A., Keefer, C., Engelhardt, A., Cooper, T., Mayer, Z., Kenkel, B., the R Core Team, Benesty, M., Lescarbeau, R., Ziem, A., Scrucca, L., Tang, Y., & Candan, C. (2017). caret: Classification and Regression Training. R package version 6.0-71. https://CRAN.R-project.org/package=caret.

Kuncheva, L. I., Bezdek, J. C., & Duin, R. P. (2001). Decision templates for multiple classifier fusion: An experimental comparison. *Pattern Recognition*, *34*, 299–314.

Lattin, J. M., Carroll, J. D., & Green, P. E. (2003). *Analyzing multivariate data*. Pacific Grove, CA: Thomson Brooks/Cole.

Li, C. S. (2011). Cluster center initialization method for K-means algorithm over data sets with two clusters. *Procedia Engineering*, *24*, 324–328.

Li, S., Harner, E. J., & Adjeroh, D. A. (2011). Random KNN feature selection: A fast and stable alternative to Random Forests. *BMC Bioinformatics*, *12*(1), 450. https://doi:10.1186/1471-2105-12-450

Man, K., Harring, J. R., Ouyang, Y., & Thomas, S. L. (2018). Response time based nonparametric Kullback–Leibler divergence measure for detecting aberrant test-taking behavior. *International Journal of Testing*, *19*, 1–23.

Marianti, S., Fox, J. P., Avetisyan, M., Veldkamp, B. P., & Tijmstra, J. (2014). Testing for aberrant behavior in response time modeling. *Journal of Educational and Behavioral Statistics*, *39*, 426–451.

McLachlan, G. J., & Peel, D. (2000). *Finite mixture models*. New York, NY: Wiley.

Meijer, R. R. (1997). Person fit and criterion-related validity: An extension of the Schmitt, Cortina, and Whitney study. *Applied Psychological Measurement*, *21*, 99–113.

Meijer, R. R., & Sijtsma, K. (2001). Methodology review: Evaluating person fit. *Applied Psychological Measurement*, *25*, 107–135.

Meijer, R. R., & Sotaridona, L. (2006). *Detection of advance item knowledge using response times in computer adaptive testing*. Newtown, PA: Law School Admission Council.

Meyer, D., Leisch, F., & Hornik, K. (2003). The support vector machine under test. *Neurocomputing*, *55*, 169–186.

Miller, A. J. (1990). *Subset selection in regression*. London, UK: Chapman and Hall.

Miller, P. J., Lubke, G. H., McArtor, D. B., & Bergeman, C. S. (2016). Finding structure in data using multivariate tree boosting. *Psychological Methods*, *21*(4), 583.

Mouselimis, L. (2018). KernelKnn: Kernel K nearest neighbors. R package version 1.0.8. Retrieved from https://CRAN.R-project.org/package=KernelKnn

Mislevy, R. (2016, November). *Simulation-based assessment: An intersection between psychometrics and data analytics*. Paper presented at the at the 16th annual meeting of the Maryland Assessment Research Center. Washington, D.C.

Pena, J. M., Lozano, J. A., & Larranaga, P. (1999). An empirical comparison of four initialization methods for the K-means algorithm. *Pattern Recognition Letters*, *20*, 1027–1040.

Primoli, V., Liassou, D., Bishop, N. S., & Nhouyvanisvong, A. (2011, April). *Erasure descriptive statistics and covariates*. Paper presented at the annual meeting of the National Council on Measurement in Education, New Orleans, LA.

Qian, H., Staniewska, D., Reckase, M., & Woo, A. (2016). Using response time to detect item preknowledge in computer-based licensure examinations. *Educational Measurement: Issues and Practice*, *35*(1), 38–47.

Qian, Y., Yao, F., & Jia, S. (2009). Band selection for hyperspectral imagery using affinity propagation. *IET Computer Vision*, *3*(4), 213–222.

R Core Team. (2014). *R: A language and environment for statistical computing (version 3.1.1)* [computer software]. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from http://www.R-project.org/

Ripley, B. (2018). tree: Classification and regression trees. R package version 1.0-37. Retrieved from https://CRAN.R-project.org/package=tree

Ripley, B., & Venables, W. (2015). class: Function for classification. R package version 7.3-14. Retrieved from https://CRAN.R-project.org/package=class

Romero, C., González, P., Ventura, S., Del Jesús, M. J., & Herrera, F. (2009). Evolutionary algorithms for subgroup discovery in e-learning: A practical application using Moodle data. *Expert Systems with Applications*, *36*, 1632–1644.

Segal, M. R. (2004). *Machine learning benchmarks and random forest regression* (UCSF: Center or Bioinformatics and Molecular Biostatistics). Retrieved from https://escholarship.org/uc/item/35×3v9t4

Saretsky, G. D. (1984). *The treatment of scores of questionable validity: The origins and development of the ETS Board of Review* (ETS Occasional Paper). Princeton, NJ: Educational Testing Service.

Sijtsma, K. (1986). A coefficient of deviant response patterns. *Kwantitative Methoden*, *7*, 131–145.

Sijtsma, K., & Meijer, R. R. (1992). A method for investigating the intersection of item response functions in Mokken's nonparametric IRT model. *Applied Psychological Measurement*, *16*, 149–157.

Sinharay, S. (2016a). An NCME instructional module on data mining methods for classification and regression. *Educational Measurement: Issues and Practice*, *35*(3), 38–54.

Sinharay, S. (2016b). Asymptotically correct standardization of person-fit statistics beyond dichotomous items. *Psychometrika*, *81*, 992–1013.

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

Sinharay, S. (2018). A new person-fit statistics for the lognormal model for response times. *Journal of Educational Measurement*, *55*, 457–476. https://doi.org/10.1111/jedm.12188

Sinharay, S., & Johnson, M. S. (2017). Three new methods for analysis of answer changes. *Educational and Psychological Measurement*, *77*(1), 54–81.

Skorupski, W. P., & Egan, K. (2011, April). *Detecting cheating through the use of hierarchical growth models*. Paper presented at he annual meeting of the National Council on Measurement in Education, New Orleans, LA.

Snijders, T. A. (2001). Asymptotic null distribution of person fit statistics with estimated person parameter. *Psychometrika*, *66*, 331–342.

Sotaridona, L. S., & Meijer, R. R. (2003). Two new statistics to detect answer copying. *Journal of Educational Measurement*, *40*, 53–69.

Steinley, D. (2006), K-means clustering: A half-century synthesis. *British Journal of Mathematical and Statistical Psychology*, *59*, 1–34.

Strobl, C., Malley, J., & Tutz, G. (2009). An introduction to recursive partitioning: Rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological Methods*, *14*, 323–348.

Tatsuoka, K. K., & Tatsuoka, M. M. (1983). Spotting erroneous rules of operation by the individual consistency index. *Journal of Educational Measurement*, *20*, 221–230.

Tendeiro, J. N., Meijer, R. R., & Niessen, A. S. M. (2016). PerFit: An R package for person-fit analysis in IRT. *Journal of Statistical Software*, *74*, 1–27.

Therneau, M.T., & Atkinson, B. (2018). rpart: Recursive Partitioning and Regression Trees. R package version 4.1-13. Retrieved from https://CRAN.R-project.org/package=rpart

Titterington, D. M., Smith, A. F., & Makov, U. E. (1985). *Statistical analysis of finite mixture distributions*. Hoboken, NJ: John Wiley.

Trabin, T. E., & Weiss, D. J. (1983). The person response curve: Fit of individuals to item response theory models. *New Horizons in Testing*, *1*, 83–108.

van der Linden, W. J. (2009). A bivariate lognormal response-time model for the detection of collusion between test takers. *Journal of Educational and Behavioral Statistics*, *34*, 378–394.

van der Linden, W. J., & Guo, F. (2008). Bayesian procedures for identifying aberrant response-time patterns in adaptive testing. *Psychometrika*, *73*, 365–384.

van der Linden, W. J., & Jeon, M. (2012). Modeling answer changes on test items. *Journal of Educational and Behavioral Statistics*, *37*(1), 180–199.

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 Krimpen-Stoop, E. M., & Meijer, R. R. (2001). CUSUM-based person-fit statistics for adaptive testing. *Journal of Educational and Behavioral Statistics*, *26*(2), 199–217.

Vapnik, V. (2005). Universal learning technology: Support vector machines. *NEC Journal of Advanced Technology*, *2*, 137–144.

Vapnik, V., & Lerner, A. J. (1963). Generalized portrait method for pattern recognition. *Automation and Remote Control*, *24*, 774–780.

Walker, A. A., Jennings, J. K., & Engelhard Jr., G. (2018). Using person response functions to investigate areas of person misfit related to item characteristics. *Educational Assessment*, *23*(1), 47–68.

Wehrens, R., & Buydens, L. M. (2007). Self-and super-organizing maps in R: The Kohonen package. *Journal of Statistical Software*, *21*, 1–19.

Weinberger, K. Q., & Saul, L. K. (2009). Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, *10*(2), 207–244.

Wollack, J. A. (1997). A nominal response model approach for detecting answer copying. *Applied Psychological Measurement*, *21*, 307–320.

Wollack, J. A., Cohen, A. S., & Eckerly, C. A. (2015). Detecting test tampering using item response theory. *Educational and Psychological Measurement*, *75*, 931–953.

Wollack, J. A., & Eckerly, C. (2017). Detecting test tampering at the group level. In G. J. Cizek & J. A. Wollack (Eds.), *Handbook of quantitative methods for detecting cheating on tests* (pp. 214–231). New York, NY: Routledge.

Wollack, J. A., & Fremer, J. J. (Eds.). (2013). *Handbook of test security*. New York, NY: Routledge.

Yan, J. (2016). som: Self-Organizing Map. R package version 0.3-5.1. Retrieved from https://CRAN.R-project.org/package=som

Zopluoglu, C. (2017). Similarity, answer copying, and aberrance: Understanding the status quo. In G. J. Cizek & J. A. Wollack (Eds.), *Handbook of quantitative methods for detecting cheating on tests* (pp. 25–46). New York, NY: Routledge.

Authors

KAIWEN MAN is at the University of Maryland, College Park, 1230 Benjamin Building, College Park, MD-20742; kman@umd.edu. His primary research interests include data mining

methods, test security, response time modeling, eye-tracking modeling, and multidimensional IRT.

JEFFREY R. HARRING is at the University of Maryland, College Park, MD, 1230 Benjamin Building, College PARK, MD-20742; harring@umd.edu. His primary research interests include finite mixture modeling, models for repeated measured data, latent variable modeling, and statistical computing.

SANDIP SINHARAY is at the Educational Testing Service, MS 12T, Princeton, NJ 08541; ssinharay@ets.org. His primary research interests include item response theory, assessment of model fit, equating, reporting of subscores, statistical methods for detecting test fraud, and Bayesian methods.