Clustering algorithm comparisons for collusion detection using mixed-type data

Soo Ingrisone, Pearson

James Ingrisone, Pearson VUE

Paper presented at the National Council of Measurement in Education (NCME) annual meeting: April 2022

Abstract

Cluster analysis is challenging using mixed-type data, which is common in testing data. The performance of five clustering methods under 12 different simulated conditions are examined and then applied to real data. This study provides guidance in selecting optimal clustering strategies for detecting aberrant examinees in the mixed-type data context.

Keywords: Innovative research that gets implemented in practice and helps promote informed decisions, Credentialing (certificates, certification, licensure), Test Security

Test security is essential for making valid interpretations from test scores and for taking actions based on those interpretations. Issues related to test security and fidelity of administration can threaten the validity and fairness of test score interpretations among examinees (AERA, APA & NCME, 2014). The advancement of the technology has enabled exams to be administered remotely. However, technological advancements have also provided examinees with an enhanced capability to collect and share test material. This results in additional challenges to maintain the security of test content, detect cheating behavior, and investigate the testing irregularities.

Test fraud is suspected when irregularities in the testing data are observed, such as, unusual response similarity among test takers, aberrant response patterns, or unexpected increases in test scores and pass rate. Studies have shown that the most common cause of irregularities prior to testing is the compromise of test content via test collusion where two or more examinees deliberately share test content or answers to items for potential gain in test performance. Examples of test collusion by examinees include accessing stolen test content posted on social media, communicating about test answers during an exam, and harvesting and sharing exam content via e-mail, website, or social media. This type of organized test security breach can have far-reaching consequences, making early detection of test security breaches essential to preventing widespread compromise (Ferrara, 2017; U.S. Dept of Ed., ISE, & NCES, 2013). Therefore, assessment programs need to be proactive about test security issues to promote confidence and ensure integrity of the test scores in the overall testing program (Olson & Fremer, 2013).

In general, justification for actions taken because of test security violations cannot be adequately supported by any one piece of evidence (Kingston & Clark, 2014). Thus, multiple

data forensic methods are suggested to corroborate findings where a group of aberrant test takers have been detected (Cizek & Wollack, 2017). According to Wollack and Maynes (2017), the detection of test collusion among multiple examinees or sets of examinees with unusual answer patterns in common have been underrepresented in the test security literature. Additionally, clustering analysis appears to be a promising methodology to detect test collusion and more research in this area has been suggested (Kim et al., 2017; Wollack & Maynes, 2017; Ingrisone & Ingrisone, 2021). Therefore, the primary purpose of this study is to explore various clustering strategies as potential test collusion detection methods and to provide advice on optimal clustering strategies.

Clustering is a fundamental technique of unsupervised machine learning. Generally, clustering is used to find distinct groups ("clusters") in unlabeled data where the observations within each group are quite similar to each other (Hastie, Tibshirani, & Friedman, 2009). However, cluster analysis is especially challenging when mixed-type data is involved, both continuous and categorical data types, which is very common in testing data. One of the reasons for this is that most of the clustering techniques are designed for a single data type, rather than mixed-type data. This commonly leads to techniques designed for a single data type being inappropriately applied to clustering mixed-type data. For example, a categorical variable is dummy coded or standardized and then the clustering technique designed for continuous data is applied to the mixed-type data. This is not an acceptable solution to the mixed-type data because it can introduce distortion in the original data and consequently may lead to increased bias (Foss & Markatou, 2018).

Similarly, some of the existing mixed-type data clustering techniques introduce challenges by requiring a user-specified weight which determines the relative contribution of the

continuous vs categorical variables. This may result in the over, or under, emphasis on the relative contribution of the categorical variables over the continuous variables in the final clusters (Ahmad & Kahn, 2017; Foss, Markatou, Ray, & Heching, 2016).

A fundamental challenge to mixed data clustering strategies arises because different data types entail a computational complexity to clustering processes. This challenge consists of simultaneously selecting the most appropriate distance metric to handle both continuous and categorical data, a clustering method to merge continuous and categorical data, and the algorithms to build optimal clusters for mixed-type data (Ahmad & Kahn, 2017). Moreover, effective strategies for clustering such data sets are surprisingly difficult to find (Foss & Markatou, 2018) and there is no clear guidance for choosing the most appropriate technique in a given context (Preud'homme et al., 2021). Therefore, this study seeks to contribute to the field by providing a deeper understanding of clustering strategies for mixed-type data and their performance for detecting test collusion. Both real and simulated scenarios are examined.

The first focus of the study is to investigate performance of various clustering strategies for detecting a collusion group using synthetic data. The following simulation conditions are manipulated: the number of candidates, the percentage of aberrant examinees in the data, and the percentage of exact match responses. The research questions are:

- Which clustering strategies are optimal for collusion group detection using mixed-type data? More specifically, which combinations of clustering method, distance metric, data transformation and optimization algorithm would perform best to detect the collusion group?
- How does the changes in the number of candidates, the percentage of aberrant examinees in the data, and the percentage of exact match responses on incorrect keys affect the

performance of the clustering strategies? Which of these factors has the most impact on clustering for detecting a collusion group?

The second focus of this study is to compare the performance of clustering strategies for detecting a collusion group using real data. The clustering techniques are applied to a data set where test collusion is suspected. About 1,000 candidates took a certification exam. The data include exam time, scaled score, and item scores. During a test administration window, a group of test items on a certification exam were wrongfully harvested and shared in a study guide. The stems, options, and keys for over 100 questions were discovered, and about 25% of the questions were found to be mis-keyed. It was suspected that the harvested content had been widely distributed as pass rates increased dramatically. The research questions are:

- Which clustering strategies are optimal for detecting organized test breach?
- What does the level of overlap agreement rate between exact response match and clustering algorithm explain?
- What could be suggested for authorized bodies to consider when contemplating disciplinary action and sanction enforcement against test security violations?

This study is divided into the two parts: In Part I, the clustering results based on various strategies are examined under 12 different simulation conditions. The similarity of the true labelling and clustering is measured and compared across clustering strategies. In Part II, the performance of various clustering techniques is compared using real data. The collusion cluster detected by various clustering strategies is compared against a potential collusion group identified by the exact response matches on incorrectly keyed items in the study guide. The level of overlap is compared to determine which clustering strategies are best to identify members of the collusion group.

Methods

In this study, three major state-of-the-art mixed data clustering methods are addressed – *partitional, hierarchical*, and *model-based* – (Ahmad & Kahn, 2017). Among the three clustering categories, five clustering strategies are compared under simulated conditions and using real data (see Table 1). Typically, mixed-type approaches implement two methods for calculating similarity: one for continuous and one for categorical data. Table 1 describes the selected clustering methods, the distance metric to merge continuous and categorical data parts and their related data transformation technique for estimating distances (similarities/ dissimilarities) that handle both continuous and categorical data, and an algorithm choice of each method. The clustering processes are carried out using the clustering algorithm packages and libraries that are readily available in the R software (R Core Team, 2020).

Table 1. Clustering strategies

Clustering Method	Merge Distance Metric	Data transformation for distance		Optimization Algorithm
		Continuous data	Categorical data	
Partitional Clustering				
Partition Around Medoids (PAM)[a]	Gower	Manhattan	Hamming/ Jaccard	Medoids
	Random Forests[e]	-	-	Medoids
K-Prototypes (KP)[b]	Weighted sum	Euclidean	Match	K-Means and K-Mode
Hierarchical Clustering				
Hierarchical Agglomerative Clustering (HAC)[c]	Gower	Manhattan	Hamming/ Jaccard	HAC + Ward link
Model-based Clustering				
KAy-means for MIxed LArge data sets (KAMILA)[d]	Ensemble-like approach	Euclidean	Probabilities	K-Means and Expectation Maximization (EM)

Note: The packages and functions in R used in the study are as follows:

a. cluster Package (pam function): https://cran.r-project.org/web/packages/cluster/index.html
b. clustMixType Package (kproto function): https://cran.r-project.org/web/packages/clustMixType/index.html
c. factoextra Package (eclust function): https://cloud.r-project.org/web/packages/factoextra/index.html
d. kamila Package (kamila function): https://cran.r-project.org/web/packages/kamila/index.html
e. randomForest Package (randomForest function): https://cran.r-project.org/web/packages/randomForest/index.html

Clustering methods

Partitional Clustering: Partition Around Medoids (PAM)

The Partition Around Medoids (PAM) is from the family of partitional clustering algorithms. Essentially, it searches for *k* representative medoids (cluster centers) in the data and then classifies each observation to the closest medoid in order to create clusters. The goal of PAM is to minimize the sum of dissimilarities between observations in a cluster and the center of the cluster (the closest medoid). It repeats the steps until the centers stop changing. Medoids are similar in concept to means or centroids, but medoids are restricted to be members of the data set. One of the advantages of PAM clustering is that it is robust to outliers compared to K-means (Kaufman & Rousseeuw, 1990). For the PAM algorithm, the distance metric can be defined by the user. For this study, both Gower distance and random forest proximity are incorporated into the PAM clustering.

Gower distance is one of the distance metrics that can handle mixed-type data. The strength of Gower's distance metric is that it is a composite measure. It computes the similarity by dividing features into two subsets, one for categorical features and the other for continuous numeric features. For continuous variables, Manhattan distance is used. For binary variables, Jaccard coefficient is used, but for categorical variables, Hamming is used to calculate the distance matrix. In essence, Gower distance is a weighted average of the distances on the different variables. It is scaled to fall between 0 and 1 (Gower, 1971).

Unsupervised Random Forests are appealing because it does not require data transformation based on types of data, thus, it preserves the original structure. When Random Forests is used in an unsupervised setting, it provides the distance metric, called proximities, that can be used for the clustering (Breiman, 2001). Also, it is robust to outliers and noise (Hastie, et

al., 2017). Proximity means the closeness between pairs of cases. The proximity between two cases is calculated by measuring the number of times that these two cases are placed in the same terminal node of the same tree of Random Forests, divided by the number of trees in the forest (Breiman, 2001). The PAM clustering with Random Forests' proximity tends to perform well in the mixed-type data (Shi & Hovath, 2006).

Partitional Clustering: K-Prototypes (KP)

The K-Prototypes clustering algorithm also belongs to the family of partitional clustering. It is a variation of the k-means algorithm which is designed to handle mixed-type data. The K-Prototypes clustering measures the distance that corresponds to the weighted sum of Euclidean distance for continuous variables and matching distance for categorical variables. If the contribution of categorical variables vanishes and only continuous variables are taken into account, it becomes k-means clustering. The steps of the algorithms are as follows: Each observation is assigned to its closest cluster prototype where cluster centers are represented by mean values for numeric features and mode values for categorical features. The cluster prototypes are updated iteratively by cluster-specific means and modes of all observations until all observations have swapped their cluster assignment or the maximum iterations have been reached (Huang, 1998).

Hierarchical Clustering: Hierarchical agglomerative clustering (HAC)

Hierarchical agglomerative clustering (HAC) creates a hierarchy of clusters organized in a bottom-up order. Starting with each observation as its own cluster, at each step the closest pair of clusters is merged as one and moves up the hierarchy until one cluster remains. It produces a tree-like visual representation called a dendrogram which represents a graphical summary of a series of joins resulting from the data clustering (Hastie, et al. 2017). Like the PAM algorithm,

the distance metric can be defined by the user. For this study, Gower distance is incorporated into the HAC clustering. The Gower distance is computed by finding the similarity between each pair of mixed data points. The aggregation method between clusters in HAC is called a linkage technique. Ward's method is selected because it is shown to cluster observations correctly in a mixed-type data context (Hummel, et al, 2017; Miyamoto, et al, 2015). Ward's method measures the proximity between two clusters in terms of the increase in the error sum of squares (SSE) that results from merging the two clusters into a single cluster. It attempts to choose the successive clustering steps so as to minimize the increase in error sum of squares at each step (Hand, et al., 2001).

Model-based Clustering: KAy-means for MIxed LArge data sets (KAMILA)

KAy-means for MIxed LArge data sets (KAMILA) algorithm is a model-based edition of k-means clustering for mixed-type data. According to Foss, Markatou, Ray, and Heching (2016), the KAMILA algorithm combines the best features of two of the most popular clustering algorithms, the k-means algorithm and Gaussian-multinomial mixture models: KAMILA does not require strong parametric assumptions for continuous data like k-means clustering does. In addition, KAMILA has attributes similar to those of Gaussian-multinomial mixture models where it balances the contribution of continuous and categorical variables without specifying weights.

In KAMILA, continuous variables are assumed to be sampled from a finite mixture distribution with spherical clusters where the density of the data is only dependent on the distance to the center of the distribution. Categorical variables are assumed to be sampled from a mixture of multinomial variables. The iterations of the KAMILA clustering consist of two steps – partition step and estimation step. The KAMILA algorithm begins with a set of centroids for

the continuous variables and a set of parameters for the categorical variables. At the partition step, the Euclidean distance with the closest centroid is computed for a set of continuous variables. This is used to estimate the mixture distribution of continuous variables. For a set of categorical variables, the probabilities of an observation within the cluster are computed. The log-likelihood of the sum of these two components, estimation of continuous variables and probabilities of categorical variables, is then used to find the most appropriate cluster for each observation. At the estimation step, the initial centroids and the initial parameters are updated to best represent the clusters. The partition and estimation steps are repeated with different initial values of centroids and parameters until the partition maximizing the sum of the best final likelihoods is obtained (Foss et al., 2016, Foss & Markatou, 2018).

Simulation

For the simulated scenarios, three variables are manipulated in the study, i.e., number of examinees, percentage of aberrant examinees, and percentage of exact response match (ERM) on incorrectly keyed items. A total of 12 simulation conditions are investigated with 100 replications across five clustering methods (see Table 2).

Table 2. Simulation conditions

Variable	Condition
Number of total examinees	500, 1,000
Percentage of aberrant examinees	10%, 20%, 30%
Percentage of exact response match (ERM) on incorrectly keyed items	70%, 80%

In order to represent realistic scenarios in the simulation, the bootstrap approach is utilized on real data. Although some distributions and models are suggested for constructing synthetic samples for the group of aberrant examinees, it is difficult to achieve a realistic picture of the collusion sample. One of advantages of using bootstrap is that it is a data-based resampling method for statistical inference without preconditions and assumptions and can be

performed in an intuitive manner without mathematics (Efron & Tibshirani, 1993). The following describes how the bootstrap samples are obtained in this study: Aberrant or collusion group samples are drawn from those examinees who had 70% or 80% exact response matches (ERM) on the set of incorrectly keyed items in the study guide and completed the exam in 120 minutes or less (Ingrisone & Ingrisone, 2021). Non-collusion examinees are sampled based on those whose exact response match is less than 50% and do not overlap with the collusion group. For instance, to make a sample size of 500 that is comprised of 10% aberrant examinees with 70% ERM, 10% of the sample of 500 (N = 50) is drawn with replacement from a group of examinees that have a 70% ERM and 90% of sample of 500 (N = 450) is drawn from the non-collusion group with replacement, respectively. The two samples are then combined to make a sample of 500. For each combined bootstrap samples, five clustering methods are employed. This process of drawing bootstrap samples and calculating the corresponding bootstrap replicates is repeated 100 times across the 12 simulation conditions. For validating clustering performance among the five clustering methods, the adjusted Rand Index (ARI) is used. The ARI measures the agreement between two partitions. The index bounds between -1 for a random partition and 1 for a perfect agreement between two partitions (Hubert & Arabie, 1985). Most of the clustering algorithms work under the assumption that the number of clusters is known in advance. This number may be either computed by other algorithms, derived from the domain, or specified by the user. In this study, clusters are specified as two because the goal of the research is to find the optimal clustering strategies for identifying collusion and non-collusion groups.

Results

Part I.

Exploratory Data Analysis

Table 1 shows score and exam time descriptive statistics for the three groups that are used in the bootstrap approach, i.e., non-collusion, collusion group with 70% ERM, and collusion group with 80% ERM. The mean scaled scores between the non-collusion group and 70% ERM groups are not significantly different. A significant mean scaled score difference is found between non-collusion group and 80% ERM group (t=5.529 *, p < .001). However, the average exam times in minutes are significantly faster for the 70% ERM and 80% ERM groups than for the non-collusion group (t=6.756[1*], p < .001 and t=15.105[*], p < .001, respectively).

Table 1. Descriptive Statistics of Score and Exam Time

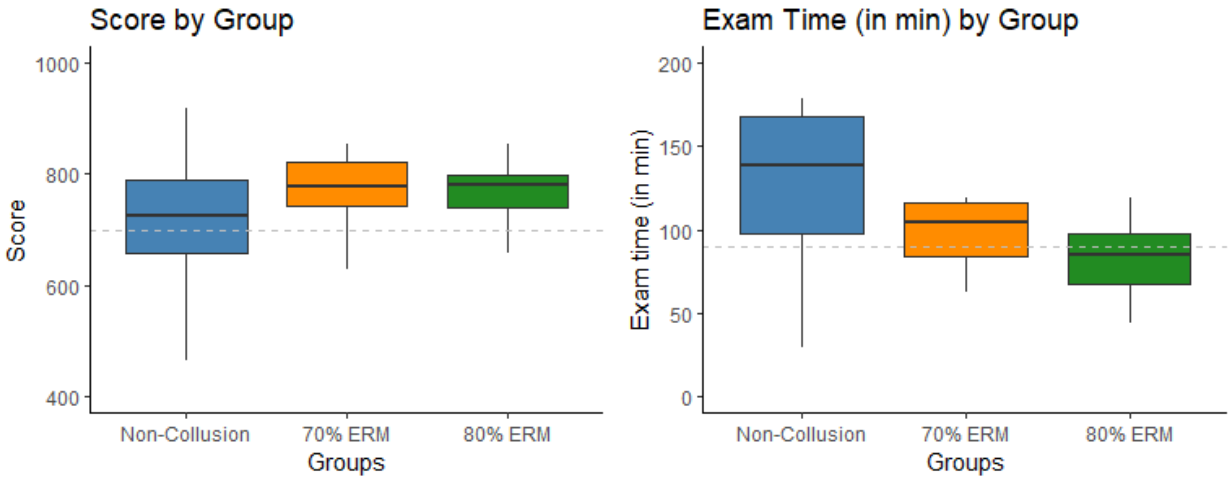
	Group	Min	25th	Median	75th	Max	Mean	SD
Score	70% ERM	545.00	742.00	776.50	821.00	854.00	761.20	86.13
	80% ERM	627.00	740.00	781.00	797.00	854.00	766.60	52.99
	Non-Collusion	424.00	659.00	724.00	789.00	918.00	720.70	94.98
Exam Time	70% ERM	62.38	83.83	104.66	116.13	119.05	98.11	19.50
	80% ERM	44.17	67.00	84.93	97.12	118.78	83.18	20.13
	Non-Collusion	29.08	98.03	138.13	168.28	179.02	131.78	38.23

Figure 1 depicts score and exam time distributions of the three groups that are being used in the bootstrap. The horizontal lines cross at 700, the scaled score cut, and 90 minutes, half the exam time, respectively. The boxplots graphically illustrate the distinct score and exam time distributions of each group. The median scores of the 70% and 80% ERM groups are substantially higher than that of the non-collusion group (776.50, 781.00, and 724.00, respectively). The score ranges of the 70% and 80% ERM groups are much narrower than the

[1] Significance at .05 α – level.

non-collusion group. Furthermore, the median exam times of the 70% and 80% ERM groups are substantially faster than that of the non-collusion group (104.66, 84.93, and 138.13, respectively). Among the three groups, the 80% ERM group has the highest median exam score and the lowest median exam time, whereas the non-collusion group has the lowest median exam score and the highest median exam time. Higher scores with narrow score ranges and faster times are typical signals for collusion group behavior.

Figure 1. Boxplot of Score and Exam Time by Group



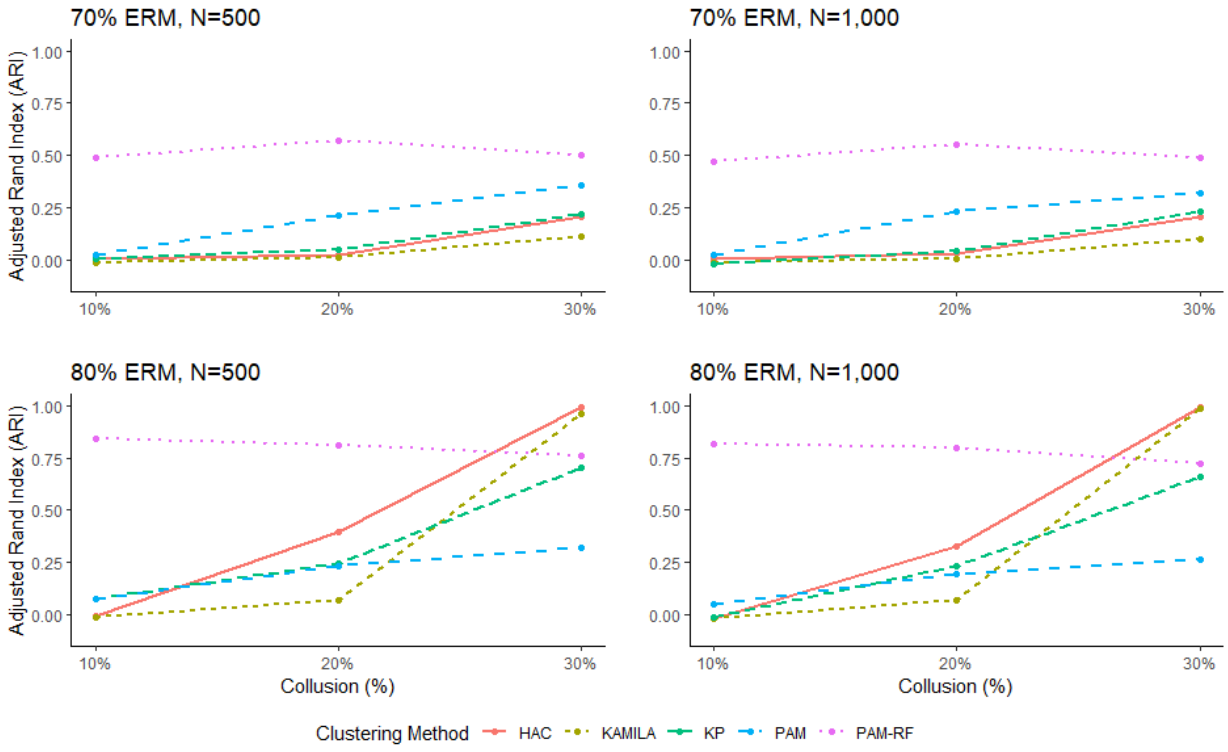
Simulation Results

Figure 2 displays the average Adjusted Rand Index (ARI) based on 100 replications across 12 simulation conditions for the five clustering methods. In general, the clustering performance (ARI) increased as the percentage of ERM increased, and the percentage of aberrant examinees increased. However, the increase in the total number of examinees did not impact the clustering performance much. Regardless of sample sizes, none of the clustering methods performed well at 70% ERM where average ARI values were less than .65, which indicates poor recovery. At 80% ERM, three of the clustering algorithms showed excellent recovery with 30% aberrant examinees where the average ARI values across all clustering algorithms were above

.73. They are HAC, PAM-RF, and KAMILA. The highest average ARI was obtained by HAC followed closely by KAMILA, .994, and .960 for N=500 and .991, and .986 for N=1,000, respectively. ARI values above .90 indicate excellent recovery. Also, the lowest ARI value at 80% ERM with 30% aberrant examinees was from PAM-RF with .757 for N=500 and .725 for N=1,000, respectively. Unlike the other clustering methods, PAM-RF presented good recovery at 80% ERM and 10% and 20% aberrant examinees with the average ARI above .80.

The results across 12 simulation conditions based on 100 replications suggest that the performance of the clustering methods varied by each condition. Overall, no clustering methods performed well with 70% ERM. At 80% ERM, PAM-RF performed well even with 10% aberrant examinees. HAC and KAMILA performed the best at 80% ERM and 30% aberrant examinees. Consequently, these two seem to be the most promising of the five clustering methods for test collusion detection in the mixed-type data context.

Figure 2. Average ARI by Clustering Method

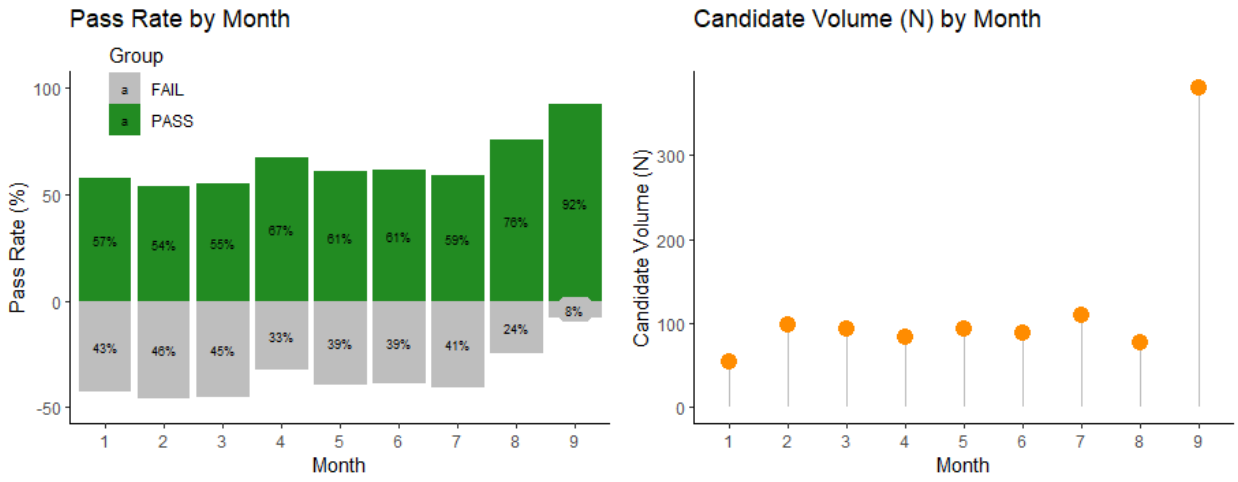


Part II.

Exploratory Data Analysis

Using real data, exploratory analysis is performed to spot any anomalies. Figure 3 displays the monthly pass rates and candidate volumes over the entire administration window. The pass rates for the first three months range from 54% to 57%. There was a noticeable increase in month 4 (67%), followed by a decline in months 5 – 7 with pass rates ranging from 59% to 62%. These pass rates averaged about a 5% increase over the first three months. A substantial jump in pass rates emerged in months 8 and 9, 76% and 92%, respectively. These are unusually high compared with the average pass rate for the previous year's administration (66%), and the current administration months 1 – 7 (approximately 60%). Also, when monthly candidate volumes were compared across months, an abnormal increase was noticed in the last month. In sum, these anomalies suggest that a group of candidates may have engaged in collusion.

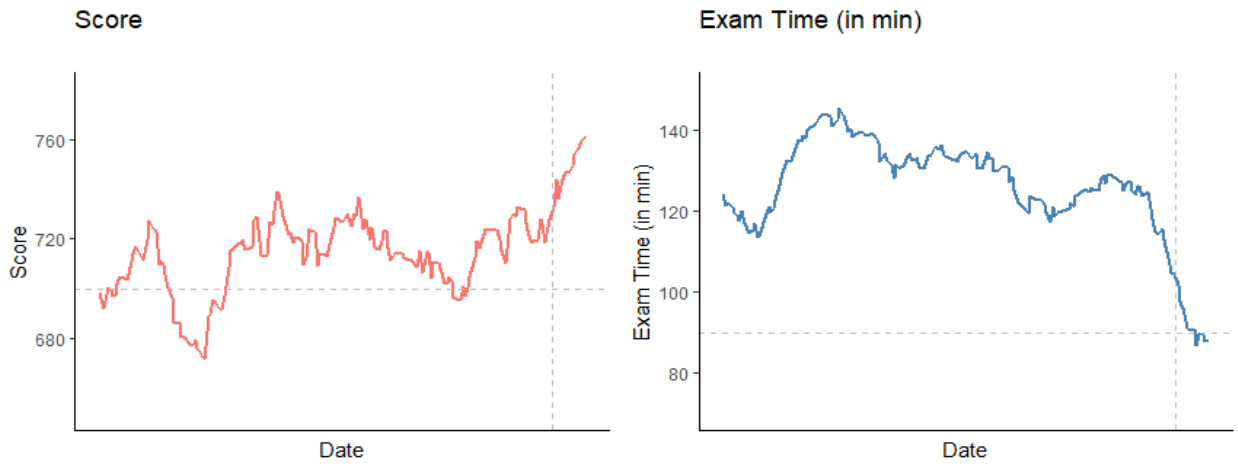
Figure 3. Pass Rate and Candidate Volume by Month



Furthermore, the moving average of scores and exam times were investigated (See Figure 4). In Figure 4, the horizontal line in the score plot refers to the passing scaled score of 700 and in the exam time plot, it indicates half of the total exam time of 90 minutes. The vertical lines in

both score and exam time plots in Figure 4 indicate the 8th month of the testing window where the unusual patterns generally appeared to emerge. The moving average of scores noticeably starts increasing, while the moving average of the exam times clearly is decreasing. These unusual scores and times are consistent with collusion behavior.

Figure 4. Moving Average of Score and Exam Time



Exam Data Results

The performance of five clustering techniques/strategies were compared using real data. The collusion clusters detected by the various clustering strategies were compared with the potential collusion groups identified by the exact response matches on incorrectly keyed items in the study guide. Table 2 shows the exact response match rate on the incorrectly keyed items from 70% to 100% and the number of candidates in each associated exact response match (ERM) group. It also includes the total overlap agreement rate between the collusion group and each ERM group as well as the overlap agreement rate for those that passed the exam. For example, the ERM group corresponding to 80% exact response match includes 356 candidates where the total overlap between the ERM and collusion is 95% agreement using KAMILA. Furthermore, 94% of those candidates passed the exam. The overlap agreement rate between the collusion group and each ERM group for KAMILA ranges from 87% to 100% for all test takers, and 94%

to 96% for those that passed the exam. The level of overlap illustrates how well the KAMILA algorithm identified members of the collusion group. It is noted that KP and PAM capture a high percentage of ERM candidates, 94% to 100%, but their pass rates range from 82% to 96%. This implies that KP and PAM may have contained non-collusion cases as well because not all ERM cases are considered collusion cases. In addition, these two clustering algorithms tend to identify more candidates that failed the exam than the other approaches. PAM-RF has the lowest agreement rate, but it identified 100% of the passing candidates across all ERM conditions.

Table 2. Overlap Agreement Rate between ERM and Clustering algorithm

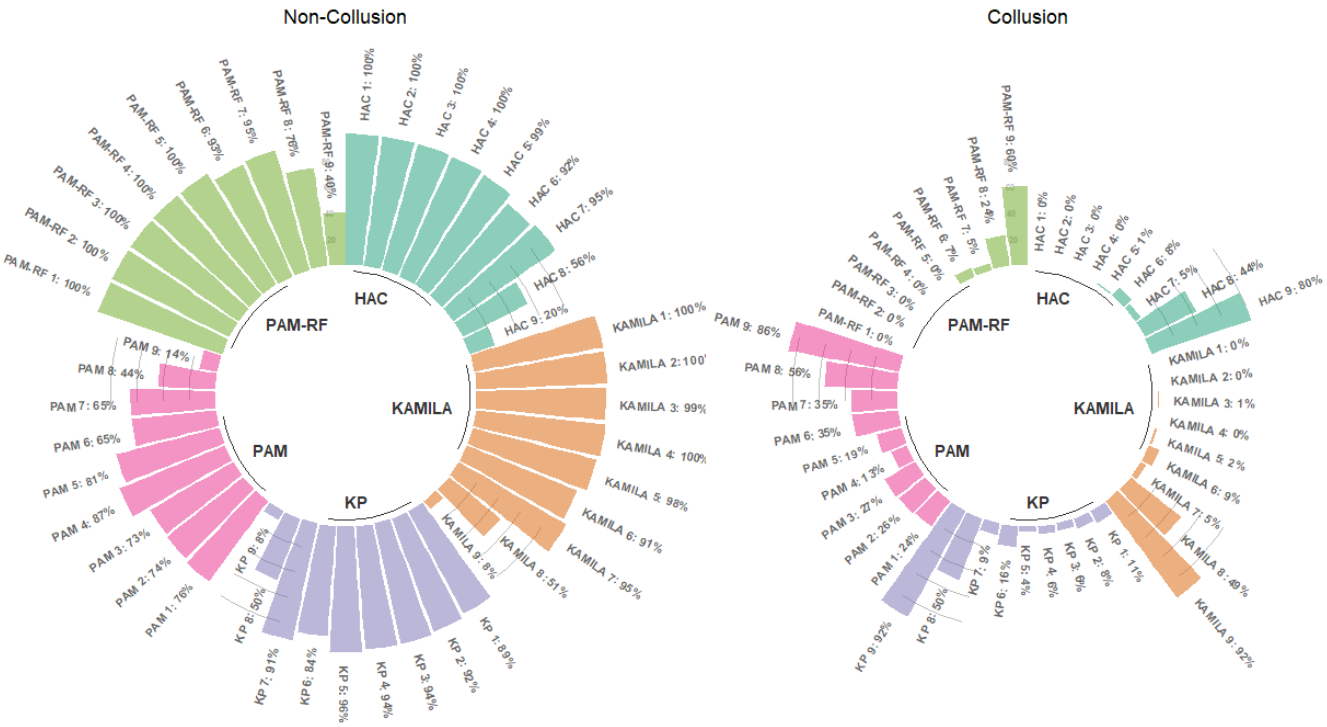
ERM		Agreement Rate (%)					Agreement Rate Pass Only (%)				
%	N	HAC	KAMILA	KP	PAM	PAM-RF	HAC	KAMILA	KP	PAM	PAM-RF
70%	400	86%	87%	94%	100%	64%	95%	94%	87%	82%	100%
75%	372	90%	91%	98%	100%	69%	95%	94%	88%	86%	100%
80%	356	94%	95%	99%	100%	72%	95%	94%	90%	89%	100%
85%	331	96%	98%	100%	100%	77%	95%	94%	92%	92%	100%
90%	298	97%	98%	100%	100%	81%	95%	94%	93%	93%	100%
95%	241	98%	99%	100%	100%	86%	96%	95%	94%	94%	100%
100%	114	99%	100%	100%	100%	89%	97%	96%	96%	96%	100%

The level of overlap was further compared to find out which of the clustering algorithms was best to identify members of the collusion group. In Figure 5, the circular bar plots show the clustering results for each clustering algorithm across 9 months. The lowest percentage of collusion candidates were detected by PAM-RF. In terms of the overlap of identified collusion, candidates identified by PAM-RF were also captured by KAMILA, HAC, KP and PAM. This means that the PAM-RF is the most conservative approach to detect collusion across the five clustering algorithms.

Figure 5 reveals that the clustering patterns based on HAC, KAMILA and PAM-RF were very similar and seemed to be consistent with the findings in the exploratory analysis where the most collusion was suspected in later months. In terms of test collusion detection by these three

clustering algorithms, KAMILA flagged more cases than HAC and PAM-RF. The collusion detected among test takers in month 9 was 60% for PAM-RF, 80% HAC and 92% KAMILA, respectively. PAM and KP detected the collusion cases in the earlier months more often than any of the other clustering algorithms. However, collusion is less likely to happen in those earlier months, thus it is rather inconsistent with the findings in exploratory analysis. In addition, those identified by PAM and KP in earlier months did not exhibit any unusual exam times. Therefore, it is concluded that PAM and KP may not provide optimal solutions to identify test collusion.

Figure 5. Circular bar plots per cluster



Among the five clustering algorithms, the collusion identified by PAM-RF, HAC and KAMILA seem to be reasonable with N = 258, 351, and 406, respectively. Among the three clustering algorithms, PAM-RF is the most conservative approach to detect test collusion and KAMILA is the most liberal approach. The difference in the number of candidates between HAC

and KAMILA detected for collusion was N = 55, where most of these testers exhibited fast exam times with high scores in months 8 and 9. This makes KAMILA a more desirable clustering strategy because it captures those with the attributes consistent with aberrant examinees.

The three Heatmaps in Figures 6 – 8 display the test collusion detected by PAM-RF, HAC and KAMILA respectively. A majority of the identified cases are overlapping across the three clustering algorithms. Also, all identified cases seem to be plausible as test collusion. Since multiple data forensic methods are suggested to corroborate findings where a group of aberrant test takers have been suspected, the clustering results of PAM-RF, HAC and KAMILA may support any actions taken as a result of test security violations.

Figure 6. Heatmap of PAM-RF Collusion Group by Score and Exam Time (N=258)

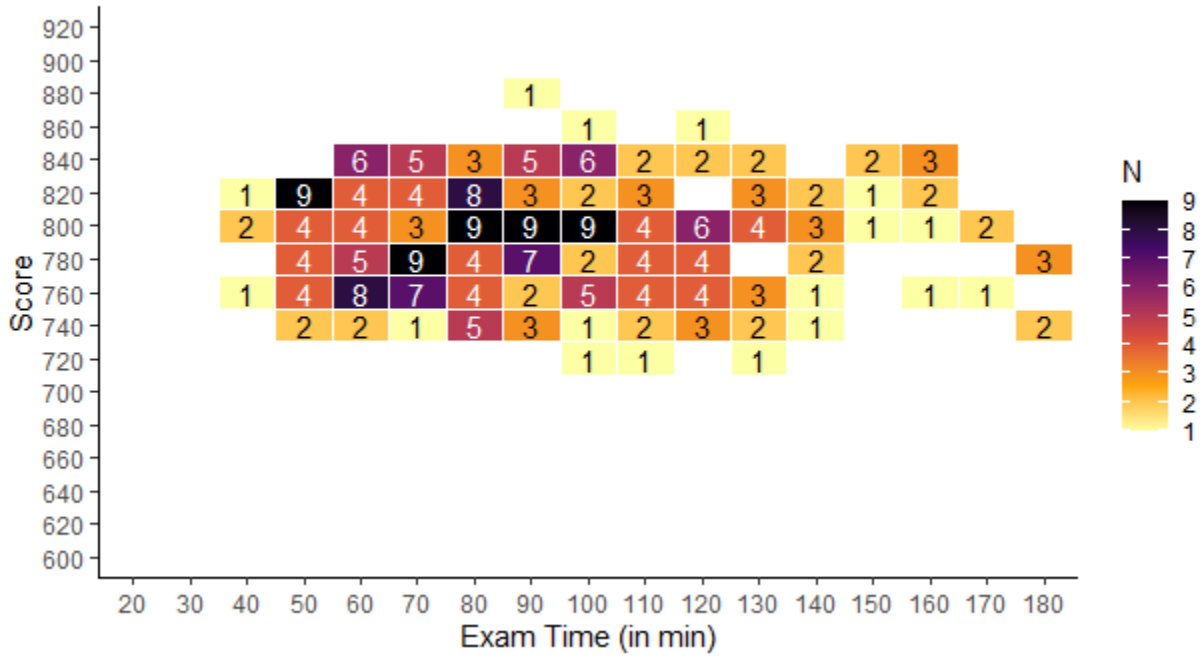


Figure 7. Heatmap of HAC Collusion Group by Score and Exam Time (N=351)

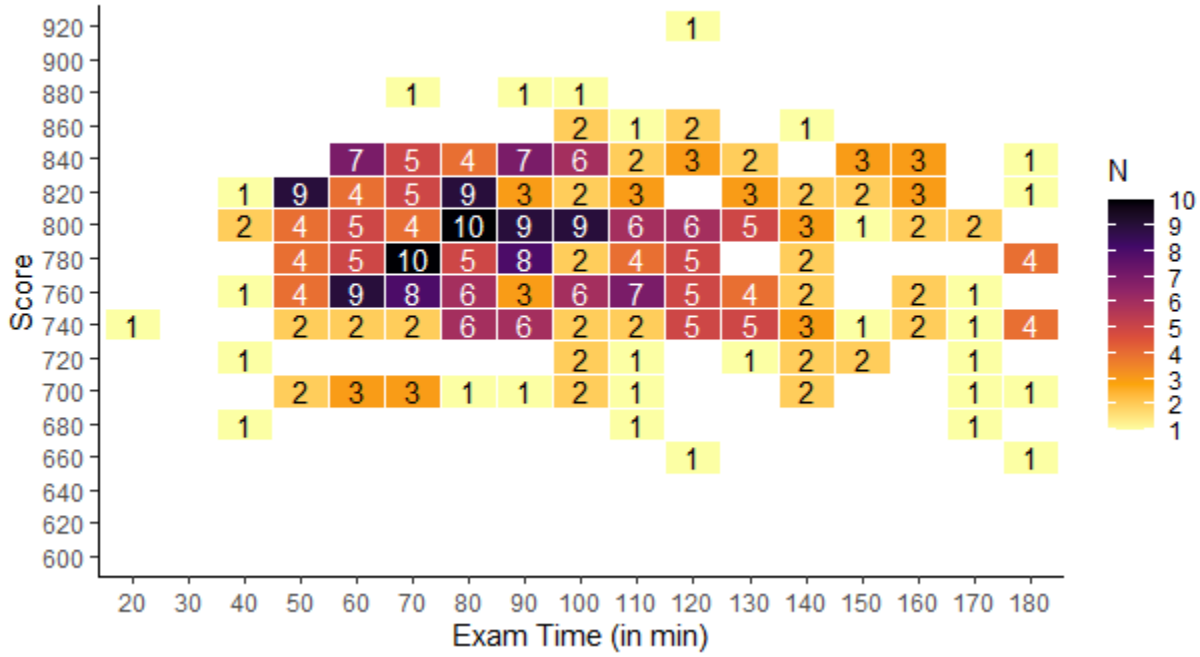
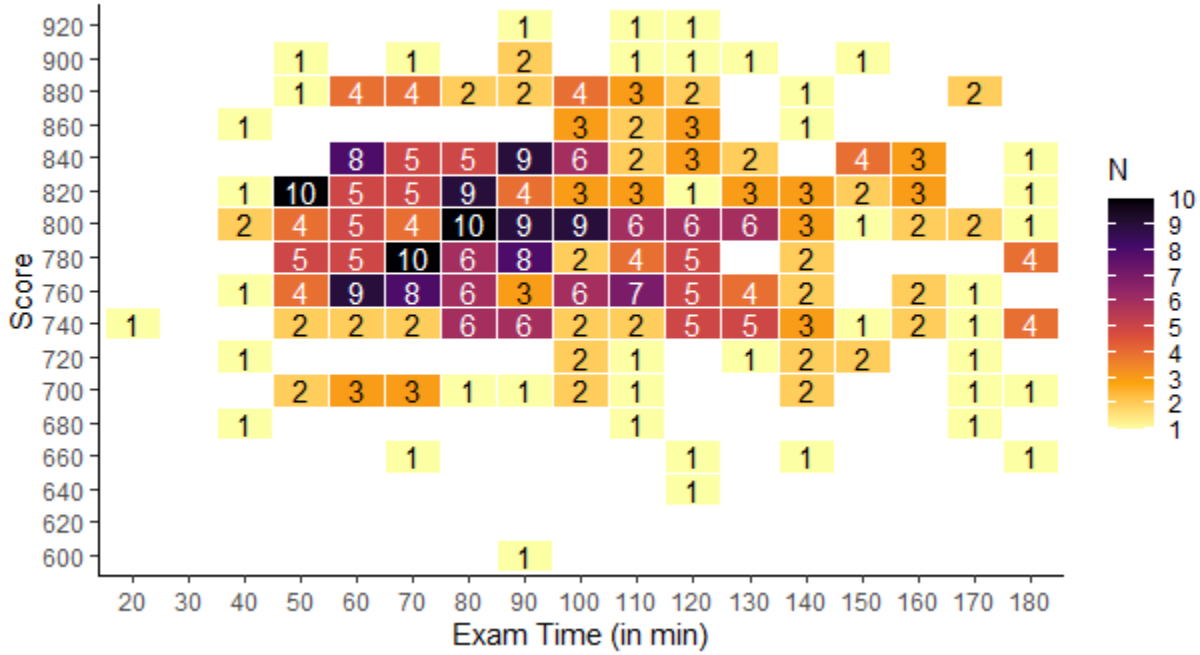


Figure 8. Heatmap of KAMILA Collusion Group by Score and Exam Time (N=406)



In summary, PAM-RF, HAC, and KAMILA provide a range of collusion group options for authorized bodies to consider when contemplating disciplinary action and sanction

enforcement. All three algorithms are promising techniques to detect test collusion and help authorized bodies resolve test security violations.

Conclusion

In general, under the simulated conditions, clustering performance improves as the percentage of aberrant examinees increase (10%, 20%, 30%), and the percentage of exact response match on incorrectly keyed items increase (70%, 80%). However, the increase in total number of examinees (n=500, n=1000) did not substantially impact the clustering performance. None of the clustering methods performed well at 70% ERM. HAC and KAMILA performed the best at 80% ERM and 30% aberrant examinees, but not as well with 10% or 20% aberrant examinees. On the other hand, PAM-RF performed well at 80% ERM with 10% and 20 % aberrant examinees, but not as well with 30% aberrant examinees.in the data. HAC and KAMILA seem to be the most promising of the five clustering methods for detecting aberrant groups in the mixed-type data context as demonstrated by an ARI above .95.

When the performance of the five clustering techniques/strategies were compared using real data, PAM-RF, HAC and KAMILA detected test collusion reasonably well. Two clustering algorithms did not perform well. PAM's performance was diminished by its tendency towards identifying all cases in the exact response match group. KP's performance may be weakened in situations where datasets contain too many categorical variables. It was also found that PAM-RF is the most conservative approach to detect test collusion and KAMILA is the most liberal approach. According to Preud'homme, et al. (2021), KAMILA is a good choice when the normal-multinomial assumption does not hold and when dealing with large datasets. PAM-RF, HAC and KAMILA all displayed good intrinsic performance for detecting test collusion and

should be considered for providing corroborating evidence to authorizing bodies when they need to take action to resolve any issues regarding test security violations.

This study contains the following limitations related to the use of bootstrap samples in simulations. Bootstrap is powerful, but it can only work with the information available in the original data. If the samples are not representative of the whole population, then bootstrap will not be very accurate, and neither will the subsequent results. Next, bootstrap can fail with distributions of small sample sizes (Efron, & Tibshrani, 1993). It is noted that the 70% ERM data was rather small and sparse leading to potentially limited bootstrap samples, possibly resulting in unsuccessful test collusion detection.

Future studies should be conducted to investigate the impact of feature selection on the performance of clustering algorithms. Although feature selection has not been employed in this study, it is considered an essential technique to reduce the dimensionality problem in the data mining task. Selecting a subset of important features for clustering help with memory, computational cost, and the accuracy of the machine learning algorithms because irrelevant or partially relevant features can negatively impact learning algorithm performance. Also, it would be easier to explain the data analysis process by selecting fewer closed and related features. Thus, it is important to explore what features are most important to detect test collusion, and which dimension reduction strategies increase the optimal performance of learning algorithms. Different evaluation measures and search techniques can be examined to produce a good feature subset in feature selection.

Test security is crucial for valid test score interpretations and fairness. This study provides practical guidelines for selecting appropriate clustering strategies to obtain optimal results for aberrant group detection.

References

Ahmad, A. & Khan, S. S. (2019). Survey of state-of-the-art mixed data clustering algorithms. *IEEE Access*, 7, 31883–31902.

American Educational Research Association (AERA), American Psychological Association (APA), and National Council on Measurement in Education (NCME) Joint Committee on Standards for Educational and Psychological Testing. (2014). *Standards for Educational and Psychological Testing*. Washington, D.C.: American Educational Research Association.

Association of Test Publishers (ATP). (2015). *Association of Test Publishers security report*: November 2015. Washington, DC: Association of Test Publishers.

Bellezza, F. S. & Bellezza, S. F. (1989). Detection of cheating on multiple-choice tests by using error-similarity analysis. *Teaching of Psychology*, 16, 151–155.

Belov, D. I. & Wollack, J. A. (2018). *Detecting groups of test takers involved in test collusion as unusually large cliques in a graph*, Law School Admission Council Research Report 18-01.

Cizek, G.J. & Wollack, J.A. (2017), *Handbook of Quantitative Methods for Detecting Cheating on Tests*, New York, NY: Routledge.

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

Efron, B. & Tibshrani,R. (1993). *An introduction to the bootstrap*, Dordrecht, Springer Science + Business Media.

Ferrara, S. (2017). A framework for policies and practices to improve test security programs:

prevention, detection, investigation, and resolution (PDIR), *Educational Measurement: Issues and Practice*, 36 (3), pp. 5-23.

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

Foss, A. H. & Markatou, M. K. (2018). Clustering mixed-type data in R and Hadoop. *Journal of Statistical Software*. 83(13), 1-44.

Foss, A., Markatou, M., Ray, B. & Heching, A (2016). A semiparametric method for clustering mixed data. *Machine Learning*, 105, 419–458.

Gower J.C. (1971). A general coefficient of similarity and some of its properties, Biometrics, 27, 857-872.

Hand, D., Manilla, H. & Smyth, P. (2001). *Principles of data mining*, The MIT press.

Hastie, T., Tibshirani, R., & Friedman, J. (2017). *The elements of statistical learning, data mining, inference and prediction* (2nd Ed.), New York, NY: Springer.

Huang Z (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2(3), 283–304.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2(1), 193–218.

Holland, P. W. (1996). *Assessing unusual agreement between the incorrect answers of two examinees using the K-index*. ETS Research Report No. 96–7, Princeton, NJ: ETS.

Hummel, M., Edelman, D., & Kopp-Schneider, A. (2017). Clustering of samples and variables with mixed-type data, *PLoS One*. 12(11).

Hunt, L., & Jorgensen, M. (2011). Clustering mixed data. *WIREs Data Mining and Knowledge Discovery*, 1, 352 – 361.

Impara, J., Kingsbury, G., Maynes, D. & Fitzgerald, C. (2005). *Detecting Cheating in Computer Adaptive Tests Using Data Forensics*, Paper presented at the 2005 Annual Meeting of the National Council on Measurement in Education and the National Association of Test Directors, Montreal, Canada

Ingrisone, S. & Ingrisone, J. (2021). *Machine learning algorithms for anomaly detection on CBT*, Paper presented at the 2021 Annual Meeting of the National Council on Measurement in Education, Virtual.

Kaufman, L., & Rousseeuw, P. (1990). *Finding groups in data*. New York, NY: Wiley.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2), 129 – 137.

Malik, A. & Tuckfield, B. (2019). Applied Unsupervised Learning with R, Packt Publishing.

Maynes, D. D. (2017). Detecting potential collusion among individual examinees using similarity analysis, In G. J. Cizek & J. A. Wollack (Eds.), *Handbook of Quantitative Methods for Detecting Cheating on Tests*, (pp. 47–69). New York, NY: Routledge.

Modha, D.S. & Spangler, W.S. (2003). Feature weighting in k-means clustering. *Machine Learning*, 52(3). 217-237.

Olson, J., & Fremer, J. (2013). *TILSA Test Security Guidebook: Preventing, Detecting, and Investigating Test Security Irregularities*. Washington, DC: CCSSO.

Preud'homme, G. et al. (2021). Head-to-head comparison of clustering methods for heterogeneous data: a simulation-driven benchmark, *Nature: Scientific Reports*, 11:4202.

R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/

Shi, T. & Horvath, S. (2006). Unsupervised Learning with Random Forest Predictors. *Journal of Computational and Graphical Statistics*. 15 (1), 118 – 138.

Szepannek, G. (2018). *clustMixType: user-friendly clustering of mixed-type data in R*, The R Journal, 10 (2).

U.S. Department of Education, Institute of Education Sciences, National Center for Education Statistics (2013). Testing integrity symposium: Issues and Recommendations for Best Practice, October 2012.

Wollack, J. A., & Maynes, D. (2017). Detection of test collusion using cluster analysis. In G. J. Cizek & J. A. Wollack (Eds.), *Handbook of quantitative methods for detecting cheating on tests* (pp. 124–150). New York, NY: Routledge.