Psychological Test and Assessment Modeling, Volume 64, 2022 (4), 425-444

# Data Augmentation in Machine Learning for Cheating Detection in Large-Scale Assessment: An Illustration with the Blending Ensemble Learning Algorithm

*Todd Zhou*[1] *& Hong Jiao*[2]

**Abstract**

Machine learning methods have been explored for cheating detection in large-scale assessment in recent years. Most of these studies analyzed item responses and response time data. Though a few studies investigated data augmentation in the feature space, data augmentation in machine learning for cheating detection is far beyond thorough investigation. This study explored data augmentation of the feature space for the blending ensemble learning at the meta-model level for cheating detection. Four anomaly detection techniques assigned outlier scores to augment the meta-model's input data in addition to the most informative features from the original dataset identified by four feature selection methods. The performance of the meta-model with data augmentation was compared with that of each base model and the meta-model without data augmentation. Based on the evaluation criteria, the best-performing meta-model with data augmentation was identified. In general, data augmentation in the blending ensemble learning for cheating detection greatly improved the accuracy of cheating detection compared with other alternative approaches.

**Keywords:** data augmentation, cheating detection, blending ensemble learning, anomaly detection algorithm

[1] Winston Churchill High School, Potomac, MD, USA

[2] University of Maryland, College Park, USA *Correspondence concerning this article should be addressed to:* Hong Jiao, Measurement, Statistics and Evaluation, Department of Human Development and Quantitative Methodology, 1230C Benjamin Building, University of Maryland, College Park, MD 20742, USA. hjiao@umd.edu

## Introduction

Test results from large-scale testing programs are often used for high-stakes decisions. When the stakes of a test are high, it is more likely that test-takers may engage in different types of cheating behaviors (Cohen & Wollack, 2006; Jurich, 2011). Cheating may often lead to spurious increase of test scores for test-takers who engage in cheating. These unethical score gains cause concerns about test security and the validity of score uses and interpretations. In addition, cheating causes person fit and item fit issues to psychometric modeling. Cheating also increases the equated scores for the test form containing compromised items and none of the state-of-the art scaling methods could mitigate this spuriously higher scores (Jurich, 2011). To assure the validity of test score uses, cheating detection is one of the psychometric analyses routinely conducted for large-scale high-stakes testing programs.

Literature has well documented different approaches to cheating detection. These include the use of person-fit statistics to identify test-takers who are misfit persons for a specific item response theory (IRT) model for test development. In the meantime, different latent variable models were proposed for modeling different cheating behaviors. For example, Shu, Henson, and Luecht (2013) utilized a gated IRT model to detect large score gain due to item exposure or item compromise by conditioning on the item exposed or unexposed status. Toton et al (2019) used conditional scaling of response time to detect examinee pre-knowledge.

Traditionally, cheating detection in large-scale assessment uses item response data and answer change data. Recently, with the prevalence of computer-based testing, more data types can be collected including process data such as response time and answer change patterns. Several studies (e.g., Man et al., 2019; Zhou & Jiao, 2022; Zopluoglu, 2019) explored item responses and response time data for cheating detection. Further, other studies (e.g., Man & Harring, 2020) included multi-modal data: assessment product data (item responses) and process data (response time) and biometric data such as visual fixation counts for cheating detection.

With increasing sources of data in computer-based tests and recent advancement in analyzing large data, machine learning methods found more applications in psychometric analyses including cheating detection. For cheating detection, machine learning is a powerful tool to recognize patterns and find similarities in answers due to plagiarism. Sahu (2016) utilized K-Nearest Neighbor (K-NN) for pattern recognition to identify the copied texts by comparing them with some existing files. Cavalcanti et al (2012) reviewed text mining methodology for cheating detection and proposed two classification models using the decision tree algorithm. To detect plagiarism which uses paraphrasing and summarization techniques, Chitra et al (2015) developed a paraphrase recognition system based on a support vector machine algorithm.

Recently, several studies explored machine learning to detect item pre-knowledge. Pan et al., (2022) proposed a machine learning approach to detecting item compromise and preknowledge in computerized adaptive testing using support vector machine and autoencoder. Zopluoglu (2019) explored Extreme Gradient Boosting in cheating detection. Man et al. (2019) investigated three supervised learning and three unsupervised learning algorithms. Zhou and Jiao (2022) explored the stacking ensemble machine learning algorithm for cheating detection. The last three studies utilized the same dataset from a large-scale high-stakes testing program with cheaters flagged based on other methods.

Zhou and Jiao (2022) study explicitly tackled the class imbalance issue via re-sampling in cheating detection where the proportion of cheaters in the data was often a very small percentage among all test-takers (reported about 3% in their study). Further, their study found that the inclusion of data augmented from the original data such as the summary statistics of item response time, the total test scores, and the number of attempts in taking the test, turned out to be effective features in cheating detection. Zopluoglu (2019) incorporated data augmentation as well by converting item responses into strings of nominal response patterns and found better performance for models including the augmented features (i.e., nominal response patterns) in terms of AUC (an increase of 0.016) and precision (an increase of 8.8% at the false positive rate of 0.01). Zhou and Jiao (2022) found that stacking learning with data augmentation improved cheating detection accuracy in terms of recall (up to 3 times), precision (up to 9 times) and F1 scores (up to 9 times). However, Zhou and Jiao (2022) augmented data only at the base model level of the ensemble stacking learning. This study intends to explore more data augmentation methods in cheating detection using the blending ensemble learning algorithm.

## Ensemble Machine Learning and Data Augmentation

### Ensemble Learning

The ensemble learning techniques include bagging, boosting, and stacking. Stacking differs from the other two ensemble learning algorithms in that it combines multiple base models to develop a meta model. The modeling process consists of two levels. The individual base models, a.k.a. heterogeneous learners, based on different machine learning algorithms, are trained separately using the training dataset and the testing dataset. Then, the meta model at the second level learns from the outputs of the first level models (Pavlyshenko, 2018). It is expected that the stacking machine learning model enhanced the prediction accuracy of a single classifier (Chan & Stolfo, 1997). Zhou and Jiao (2022) demonstrated the application of the stacking learning algorithm for cheating detection in large-scale assessment and found the stacking learning with data augmentation improved cheating detection accuracy in general. The number of test-taking attempts and the summative information of response time and the total test scores were added into the feature space for augmentation at the base-model level.

Blending (Khyani et al, 2021), an ensemble machine learning method, is derived from stacked generalization. With the same structure of stacking learning, the blending learning method simplifies the stacking process by using one split of training and testing datasets to develop a meta-model using base models' prediction results. Different from the stacking ensemble learning using k-fold cross validation, the blending ensemble learning splits the original data into two parts: a training and testing dataset respectively. Then, the training set is further split into two datasets for training the base models (training 1) and a meta-model (training 2) respectively. Training dataset 1 is used to train the base models. The predicted results from each base model are used as the features in training the meta-model using Training dataset 2. Finally, the developed meta-model makes predictions on the testing dataset. In sum, while the stacking trains the meta-model using out-of-fold predictions from base models via k-fold cross validation, the blending fits the meta-model with the predictions generated from a holdout dataset. The blending learning is a stacking method using a more straightforward way to set up the dataset for the meta-model training (Brownlee, 2020).

## Data Augmentation

Data augmentation is expected to improve the performance of machine learning models and increase a machine learning model's generalization by creating variability in data. Two types of data augmentation are often implemented in machine learning: one augments the sample space by increasing more synthetic subjects while the other augments the feature space by increasing the number of features. Data augmentation is commonly applied in adding more samples for machine learning-based aberrant detection such as cancer detection and cheating detection, by creating more synthetic data points. The Synthetic Minority Oversampling Technique (SMOTE) is a data augmentation approach in machine learning. For example, Zhou and Jiao (2022) applied augmentation in increasing the samples by using the SMOTE technique to generate synthetic cheater data based on the KNN algorithm to tackle the class imbalance issue. Tiong and Lee (2021) applied a data augmentation approach to add 60 data samples representing the aberrant behavior class for online exam cheating detection. Alzubaidi et al (2020) augmented their breast cancer training images by utilizing image processing techniques including rotating, flipping and adjusting brightness. All these studies augmented the data by adding more samples to the original dataset.

The representation of the feature vector significantly impacts the performance for most machine learning models (Heaton, 2020). The lack of feature representation in training data sometimes leads to worse prediction results than the ones caused by the lack of data samples. Heaton (2020) indicated that adding new features based on the other features could augment the feature representation of the input data, to improve the model performance. Data augmentation can be implemented by augmenting the feature space. Micenková et al. (2015) used outlier scoring functions to create outlier scores as transformed features to build a richer data feature representation for learning. Devries and Taylor (2017) performed the transformation of the existing data vector in

the learned feature space, to augment the dataset. They augmented the feature space by encoding the sequence, extracting the context vector, and performing the transformation, such as adding noise, into the feature space. Zopluoglu (2019) augmented the feature space by converting item responses into a string of nominal response pattern. Zhou and Jiao (2022) augmented the data at the base-model level in the stacking learning algorithm application by adding the number of test-taking attempts and the summary statistics into the feature space. In addition to data augmentation via transformation of existing features, image feature space can be augmented by different brightness, color, hue, orientation and cropping, while the audio features can be augmented by changing the pitch, timbre, loudness, spatial location, and other spectrotemporal features.

This study explored multiple unsupervised outlier detection algorithms to estimate outlier scores which augment the data for the second-level, the meta-model development in blending learning to increase learned representations of the original data (Micenková et al., 2014). Zhou and Jiao (2022) demonstrated the use of the stacking learning to develop meta-models only using the prediction outputs from the base models without retaining the characteristics of the original dataset. This study further explored the effectiveness of data augmentation via adding the outputs from outlier detection methods and the selection of the most effective features from the original dataset to the meta-model development.

## Data Augmentation in the Meta-Model Development in the Blending Ensemble Learning

### Blending Ensemble Learning

To illustrate the proposed data augmentation method in cheating detection using machine learning, the blending ensemble learning algorithm was used and illustrated in Figure 1. At the first level of the blending learning, the base models were developed in the same manner as those used in Zhou and Jiao (2022). At the second level, meta-models were developed based on the prediction outputs from the base models and two types of data augmentation: outlier measures from anomaly detection and the selected most effective features from the original feature space.

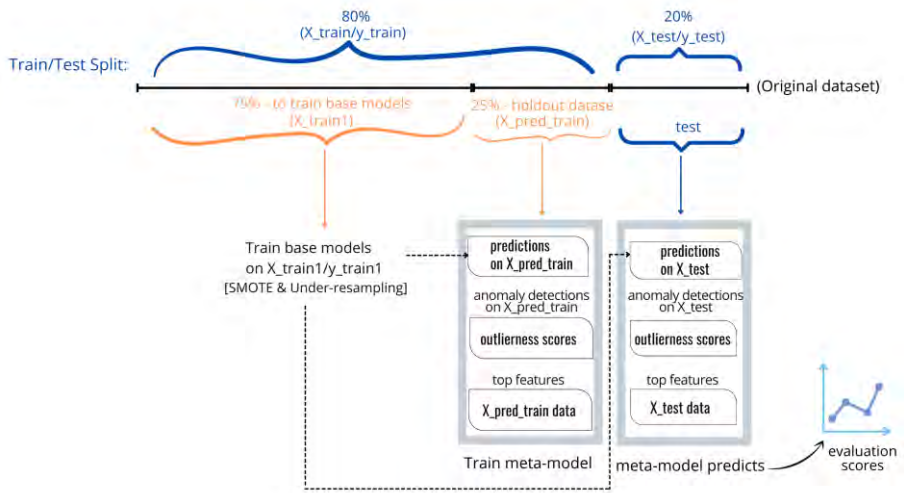


***Figure 1.***

Blending ensemble learning.

## Data Augmentation

This study explored different methods to increase the feature representation in the cheating detection dataset for the meta-model development in the blending ensemble learning from the following two aspects. First, the feature space was expanded by including the outlier measures from different anomaly detection methods showing the aberrant item responding behaviors. Further, the most effective features from the original data which represent the raw data characteristics were incorporated in the feature space in the meta-model development. The proposed data augmentation scheme is illustrated in Figure 2. In general, the meta-model development with data augmentation consists of three components as graphed in the purple dot rectangle at the lower part of the figure.

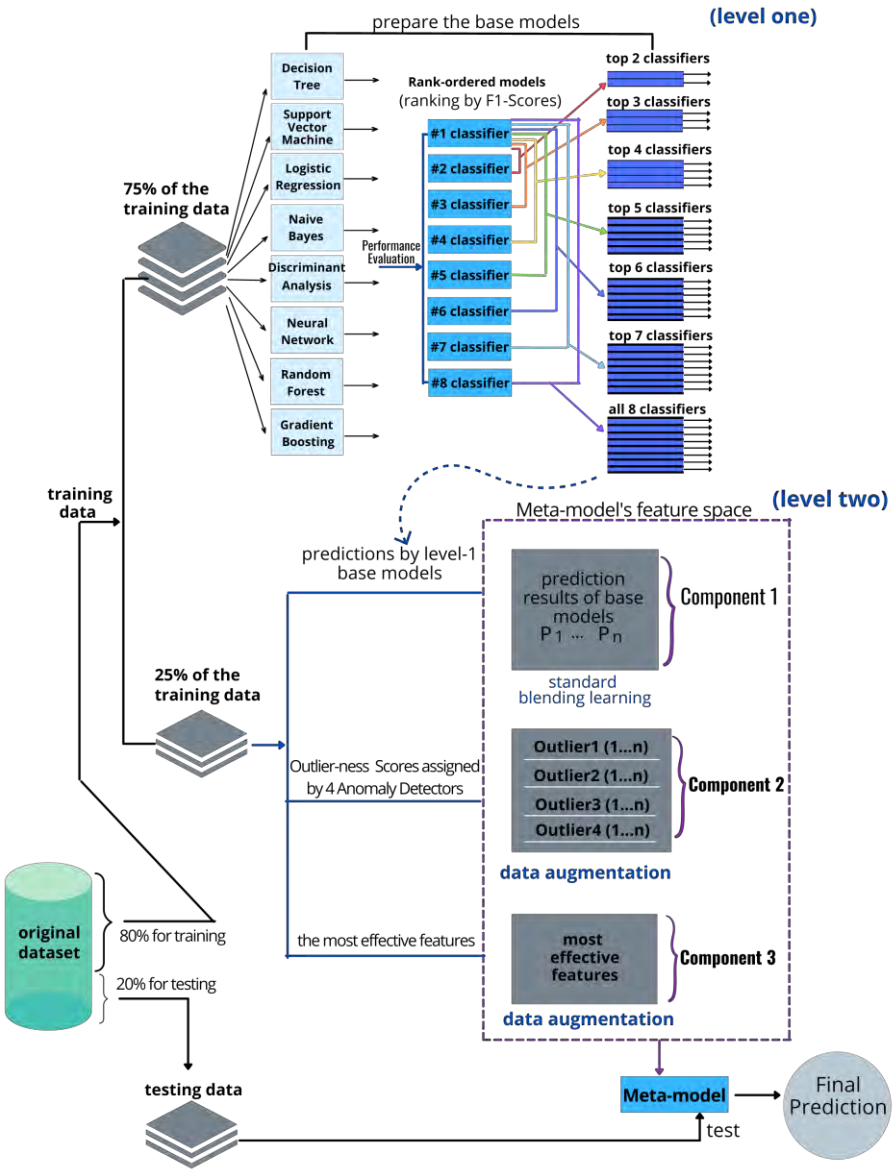


***Figure 2.***

Data augmentation in blending learning

## Component 1. Base model development

For the base model development, eight models were explored including six non-ensemble learning models and two ensemble learning models via the Scikit-learn (sklearn) Machine Learning library in Python (Pedregosa et al., 2011). The six non-ensemble algorithms were Decision Tree (DT), Naïve Bayes (NB), Neural Network (NN), Discriminant Analysis (DA), Logistic Regression (LR) and Support Vector Machine (SVM) while the two ensemble learning algorithms were Random Forest (RF) and Gradient Boosting (GB). Resampling was applied on the training dataset to tackle the class imbalance issue. The performances of all eight classifiers were rank-ordered. Different sets of base models' outputs were used as part of the input data to build the meta-model.

## Component 2 - Data augmentation via anomaly detection techniques

Outlier indices via anomaly detection methods were used to enlarge the feature space of the meta-model for data augmentation revealing the degree of anomalousness of each test-taker. This study explored four anomaly detection algorithms to capture different forms of outliers including Isolation Forest, Elliptic Envelope, One Class SVM, and DBSCAN given these four algorithms are commonly used in anomaly detection methods (Jose, 2019).

Isolation Forest is similar to Random Forest in that both of them are built upon an ensemble of binary decision trees, but Isolation Forest explicitly identifies anomalies instead of normal observations. Isolation Forest algorithm randomly picks a feature and a value of that feature, comparing it with the minimum or maximum value of the feature value, and repeats the process till the observation has no values falling in the range of that feature in the dataset. The number of times to go through to reach the isolation reflects the outlier-ness.

The Elliptic Envelope method views all features as a whole and applies a multivariate Gaussian distribution to the dataset. An ellipse is marked according to the gaussian distribution of data samples. The outlier-ness values are assigned to each data point. Those data points that lie outside the ellipse are considered outliers.

In general, SVM finds a max-margin hyperplane to separate different classes. Schőlkopf et al. (1999) extended the SVM methodology to One Class SVM, also called unary class-modeling, by finding a hypersphere using only one particular class' information to differentiate that class from all other classes. It minimizes the hypersphere of the single class and considers other data points outside the hypersphere to be outliers.

DBSCAN stands for Density-Based Spatial Clustering of Applications with Noise. As a density-based clustering algorithm, it observes the core of high density in data

points, expands the core data into clusters by gathering nearby neighbors, and locates the outliers which exist in low density with their nearest neighbors far away.

The outputs, the outlier score of each data point, from each of these outlier detectors described above were added into the feature space of the meta-model, to improve the data representation and augment the feature space.

### Component 3 - Data augmentation via top feature selection from the original dataset

This study also incorporated selected original features in the meta-model development to augment the facets of learning perspectives. Miao and Niu (2016)'s experiment shows the model performance of learning can be improved by adopting original feature selection. To reduce the noise and insignificant, redundant characteristics affecting the model performance and interpretability, feature selection methods were used to improve prediction accuracy and model performance (Chen et al., 2020). A subset of effective features was selected from the full feature set. In addition to the augmented features including the attempts of taking the test and the summary statistics included in Zhou and Jiao (2022), the most effective features were selected via four feature selection algorithms.

The four feature selection methods include two filter methods: Chi-square statistics and Pearson Product Moment Correlation (PPMC), one wrapper method: recursive feature elimination (RFE) and one embedded method: extremely randomized trees or Extra Trees Classifier. Chi-square statistics calculates the difference between the real values and the expected values based on frequencies for categorical features (McHugh, 2013). Chi-square $x^2$ test of independence is used to measure the relationship between two categorical variables. The SelectKBest function in the feature selection module of sklearn was applied with the score function based on $x^2$ test to select the best features from item responses. The PPMC was computed to identify the most effective features among response time and the summative statistical data features which are continuous variables. Recursive Feature Elimination (RFE) in sklearn fits the model multiple times to measure the feature importance by removing less important features in each iteration. The process continues until a defined threshold is reached. The decision tree algorithm was used as RFE's backend estimator which provides the relative importance of features. The advantage of using decision tree algorithm is due to its straightforward learning interpretation. It is easy to compute how much the feature contributes to the result and derive their importance on decision-making. Similar to Random Forest, extremely randomized trees or Extra Trees Classifier ensembles the results of de-correlated decision trees to generate its classification outputs (Geurts et al., 2006). Gini Index used in building the forest is computed as the importance of the feature. The Gini Importance Index ranks the features and selects the top features.

In the end, the most effective features were selected from the original feature space for the base model development and incorporated as one component in data augmentation in the meta-model development. The features selected by multiple methods were included only once in the meta-model development.

## Method

To investigate the impact of data augmentation in blending ensemble learning for cheating detection, an empirical study was conducted to evaluate the performance of the developed meta-model with data augmentation. The prediction accuracy was compared with that of the blending model developed without data augmentation for the meta-model and those of the eight base models for cheating detection. More specifically, different models with data augmentation were developed using different sets of augmented datasets including augmented outlier measures only, selected most effective features only, and both. All analyses were conducted in Python using Google Colaboratory or "Colab". Google Colab provides Tesla K80 GPU and its cloud service provides 4992 CUDA cores and a memory bandwidth of 480GB/sec (240GB/sec per GPU).

### Data

The dataset used in this study was from a large-scale licensure test with likely cheating cases flagged (Cizek & Wollack, 2017). The test form consists of 170 dichotomous items. Among the 1636 test-takers, 46 were flagged as likely cheaters. The dataset consists of item responses (iraw), response time (idur) and the number of attempts of taking the test. In addition, the total test score, the mean, median, minimum, maximum, and the total item response time across 170 items were added into the feature space for every test-taker. In total, 347 variables were used for each base model development.

As only a very small percentage of test-takers were flagged as likely cheaters, 2.81% of the 1636 test-takers, the class imbalance issue was tackled using dual resampling, i.e., the combination of Synthetic Minority Oversampling Technique (SMOTE) proposed by Chawla et al. (2002) and randomly under-sampling applied on the training dataset only (Zhou & Jiao, 2022). For illustration, a rate of 0.4 for the SMOTE oversampling and a rate of 0.5 for the RandomUndersampling were adopted.

## Base model and meta-model development

To implement the blending ensemble learning algorithm, the original data was split into training and testing datasets with 20% of the original sample for testing (X_test/y_test), and 80% sample (X_train/y_train) as the training set. For the 80% training data, 75% of the training set (X_train1/y_train1) was used to train the eight base models while 25% of the training set as the holdout (X_pred_train) for base models to generate prediction results. The eight base models were implemented using the Scikit-learn (sklearn) Machine Learning library in Python (Pedregosa et al., 2011). The eight base models were rank-ordered to obtain the best set of base models for the meta-model development.

The predicted results for each top ranked base model were used to develop a meta-model based on the 25% training data in addition to the other two augmented data/feature components, namely different measures of outliers and the most effective features. The resampling methods were applied on the base model's training data, X_train1/y_train1. The model performance was evaluated on the testing dataset, the X_test/y_test, which is 20% of the original dataset. For the meta-model development using gradient boosting, anomaly detection algorithms and the automated feature selection methods were implemented using the Scikit-learn and pandas in Python.

## Evaluation criteria

Due to the imbalanced class sizes in the cheater and non-cheater groups, overall accuracy defined in terms of the percentage of correct classification is not an appropriate evaluation criterion. In this study, the F1-score, the harmonic mean of recall and precision, was adopted for model evaluation (Forman & Scholz, 2010). In addition, both Recall and Precision were reported as well. Recall (Sensitivity) evaluates a model's ability to predict true positives (cheaters) out of the total actual positives while Precision measures how a model performs at detecting true cheaters out of those predicted as cheaters. Further, the False Positive rate was reported to measure how many non-cheaters were misclassified as cheaters. Recall, Precision, F1-score, and False Positive rate are computed as presented in equations 1, 2, 3 and 4.

$$Recall = \frac{True\ Positive}{TruePositive+FalseNegative}. \quad (1)$$

$$Precision = \frac{True\ Positive}{TruePositive+FalsePositive}. \quad (2)$$

$$F1\ score\ = \frac{2*Precision*Recall}{Precision+Recall} = \frac{2*TruePositive}{2*TruePositive+FalsePositive+FalseNegative}. \quad (3)$$

$$False\ Positive\ rate\ = \frac{False\ Positive}{False\ Positive + True\ Negative}. \quad (4)$$

## Results

This section summarizes the results of the study using data augmentation in developing a meta-model for blending ensemble learning. First, the base model performance is summarized. Then, a list of top features selected by different methods are presented. Further, the outlier measures from each anomaly detection method are summarized. Finally, the meta-model performance with and without data augmentation are compared with the base models in terms of the evaluation criteria.

### Base Model Development

Eight base machine learning models were trained with a SMOTE rate of 0.4 and a RandomUndersampling rate of 0.5 to deal with class imbalance. Figure 3 summarizes the model performance. F1-scores were used to rank order the model performance. Among the eight base models, Gradient Boosting performed the best with the highest F1-score (0.4015), the third highest Recall (0.2939), the second highest Precision (0.6964), and the second lowest False Positive rate (0.0050). Random Forest was ranked as the second best with a F1-score of 0.3402, a Recall of 0.2248, the highest Precision of 0.7917 and the lowest FPR of 0.0025. Support Vector Machine had the worst performance.

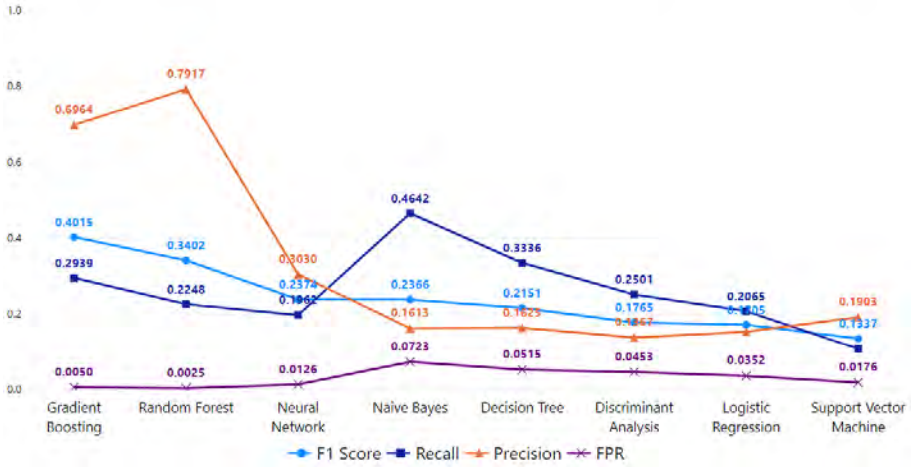


***Figure 3.***

Performance of base models at level one.

## Data Augmentation for the Meta-Model Development

The top10 effective features are summarized in Table 1. Each feature selection algorithm selected ten most effective features, except that the Pearson correlation coefficients selected the nine most effective features. The selection of 9 effective feature based on the Pearson correlation is due to the correlation score limit being set as 0.1. Schober et al. (2018) stated that most researchers agree that a coefficient of <0.1 indicates a negligible relationship. Thus, the variables with a correlation coefficient less than 0.1 were not considered in this study under this criterion. Table 1 also presents the values from each of the three algorithms. RFE function only returns the rankings without values. Thus, n/a is shown in Table 1. A total of 39 features were selected. With the removal of the overlapping features, 33 unique features were retained.

***Table 1:***
The Top-10 Features Based on the Feature Selection Algorithms

| Rank | Chi-Square Statistics | | Pearson Correlation | | RFE | | Extra Trees Classifier | |
|---|---|---|---|---|---|---|---|---|
| | Feature | Statistics | Feature | Statistics | Features | Statistics | Features | Statistics |
| 1 | iraw.32 | 4.698 | *mean.idur* | 0.177 | *mean.idur* | n/a | *tot.time* | 0.0244 |
| 2 | iraw.77 | 3.716 | *tot.time* | 0.174 | idur.60 | n/a | *mean.idur* | 0.0240 |
| 3 | iraw.147 | 2.995 | *median.idur* | 0.169 | *iraw.tot.score* | n/a | *median.idur* | 0.0169 |
| 4 | iraw.57 | 2.951 | idur.162 | 0.132 | idur.71 | n/a | *idur.146* | 0.0102 |
| 5 | iraw.86 | 2.808 | idur.67 | 0.118 | idur.133 | n/a | *max.idur* | 0.0082 |
| 6 | iraw.152 | 2.762 | idur.38 | 0.107 | idur.88 | n/a | idur.36 | 0.0080 |
| 7 | iraw.12 | 2.708 | idur.96 | 0.104 | idur.59 | n/a | idur.114 | 0.0078 |
| 8 | iraw.127 | 2.699 | idur.81 | 0.102 | *idur.146* | n/a | idur.84 | 0.0076 |
| 9 | iraw.140 | 2.424 | min.idur | 0.100 | *max.idur* | n/a | *iraw.tot.score* | 0.0076 |
| 10 | iraw.96 | 2.535 | n/a | | idur.69 | n/a | idur.142 | 0.0068 |

***Note:*** Chi-square statistics on item response; Pearson correlation on item response time and summary statistics; RFE on all features; Extra Tree Classifier on all features.

The four outlier detection methods generated outlier scores for each test-taker, including Isolation Forest, Elliptic Envelope, and One Class SVM with values of 1 and -1, and DBSCAN with values of 0 and -1. Four columns holding these scores, Outlier1, Outlier2, Outlier3 and Outlier4, were added into the learning space. Table 2 summarizes the number of test-takers who were flagged as outliers based on each anomaly detection method. Different anomaly detection methods flagged different numbers of test-takers as outliers.

***Table 2:***

Summary Statistics of the Outlier Scores from Each of the Outlier Detection Methods.

| Methods | Number of inliers | Number of Outliers |
|---|---|---|
| Isolation Forest | 292 | 36 |
| Elliptic Envelope | 295 | 33 |
| One Class SVM | 313 | 15 |
| DBSCAN | 308 | 20 |

## Meta-Model Development

To develop a meta-model, the prediction outcomes from each base model were used as one component of input features. Based on the ranking of the 8 models (GB, RF, NN, NB, DT, DA, LR, SVM) presented in Figure 3 from the best to the worst in terms of the F1 scores, seven base-model sets containing different numbers of the top models were used for the meta-model development with Blending2 containing the top 2 base models while Blending8 consisting of all 8 models. Take Blending6 as an example, it included the top-6 base models: GB, RF, NN, NB, DT, DA to develop a meta-model.

The meta-model was developed based on 25% of the training data with the input features from the three components: the predicted probabilities from each of the level-one base models, outlier measures from each anomaly detection method, and values from the most effective features selected from different effective feature selection methods. Taking blending6 as an example, a total of 43 features, namely 6 predicted probabilities, 4 outlier measures, and 33 most effective features were used for the meta-model development.

As shown in Figure 4, all models are ranked in terms of the F1-scores based on the predicted results from each base model plus the two types of data augmentation: outlier measures and the selected top effective features. The best-performing meta-model is the one blended based on the top six base models of GB, RF, NN, NB, DT and DA. This best meta-model yielded the highest F1-score (0.7857), Recall (0.7333), and Precision (0.8462), and one of the lowest False Positive rates (0.0064). These evaluation criteria turned out to be the best among studies using the same dataset for cheating detection including Zopluoglu (2019) and Zhou and Jiao (2022), where the latter reported the highest F1 score of 0.586. Among the 7 blending sets, 6 of them performed better in terms of the F1 score than that reported in Zhou and Jiao (2022).

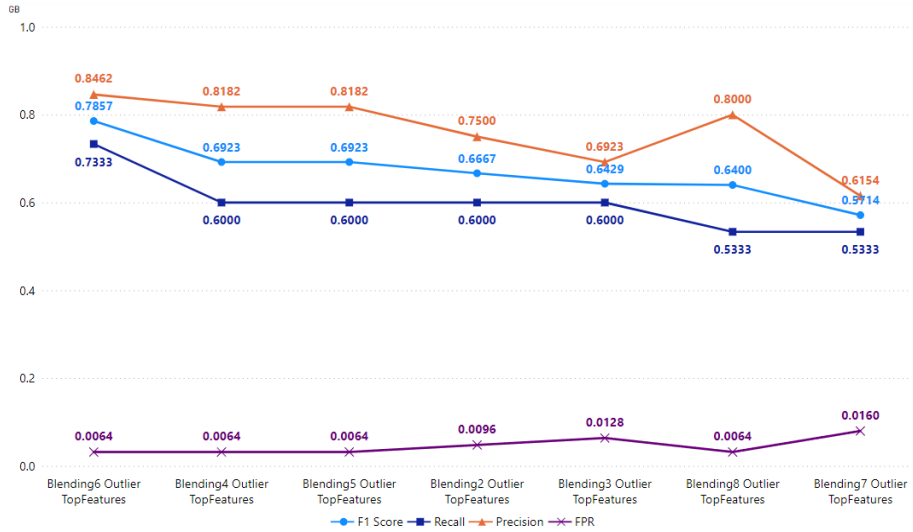


***Figure 4.***

Model performance comparison of seven Blending models with data augmentation.

To compare the best performing meta-model based on blending with other competing models, Figure 5 presents the performance of the best blended meta-model with two types of augmented data and those of eight base machine learning models. Further, the performance of the meta-models with one type of augmented data is also included in the figure. In general, the meta-model based on blending incorporating two types of augmented data performed the best with the highest F1-score, balanced Recall and Precision, both are larger than 0.5, and also a relatively low False Positive rate. The two meta models based on either augmented data of outlier measures or the selective effective top features performed better than the blending learning without data augmentation. Compared with the base models without blending, the meta-models developed based on blending all performed much better than each base model including the two other ensemble learning: Gradient Boosting and Random Forest.

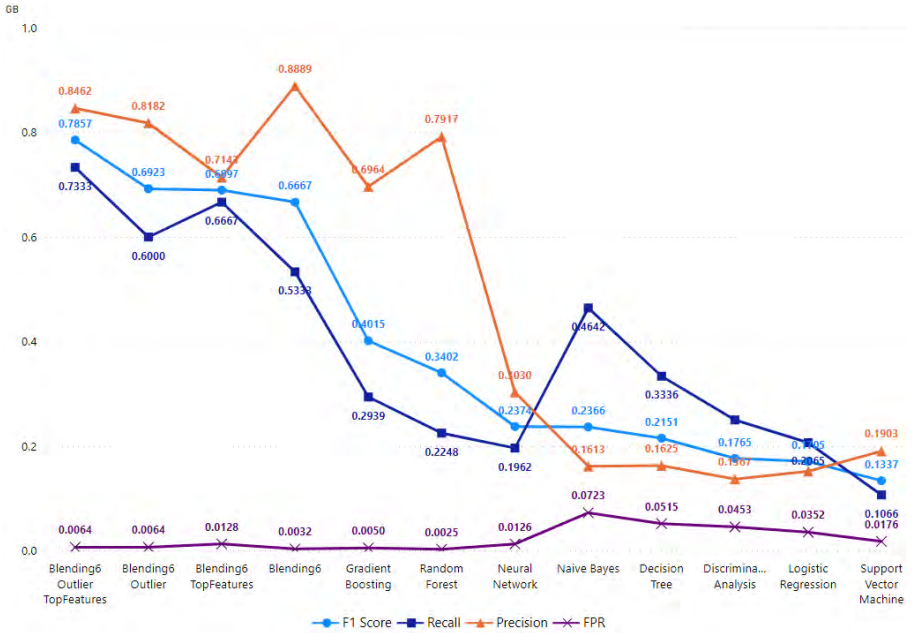


***Figure 5.***

Model performance comparison among the meta-models with and without data augmentation and the base models.

## Summary and Discussion

Machine learning algorithms have been explored for different psychometric analyses. Cheating detecting using machine learning is such an application. Though previous studies investigated different supervised, unsupervised, ensemble or non-ensemble machine learning algorithms for cheating detection, this study explored the utility of data augmentation based on anomaly detection methods and the selection of the most effective features in developing meta-models in the blending ensemble learning algorithm for cheating detection. This empirical study analyzed the dataset used in other similar studies and found that the meta-model with two types of data augmentation proposed in this study performed better than the blending learning models with one or none augmented data. Furthermore, these meta-models based on blending ensemble learning performed much better than any of the eight base machine learning models to detect cheaters in a large-scale assessment.

This study explored two types of data augmentation in expanding the feature space. Other data augmentation methods can be explored in future studies. For example, psychometric analysis results such as person fit measures can be used as a type of data

augmentation. In addition to augmenting the feature space, future studies can augment the data for cheating detection by generating more synthetic cases based on the available data to increase the sample size for the cheater group, often with an extremely low proportion of the test-takers. Though Zhou and Jiao (2022) explored such data augmentation in oversampling to deal with the class imbalance issue using the SMOTE procedure, this type of data augmentation in the sample space warrants more extensive investigation. Integrating both perspectives of data augmentation in both the feature space and the sample space are worthy of further exploration.

Though the ensemble learning algorithms such as blending and stacking have been demonstrated as effective learning algorithms in cheating detection in large-scale assessment, the interpretability of the results might be challenging due to the integration of information at different layers in the model development. Future research can explore this issue and explore the interpretability of the results.

## Reference

Alzubaidi, L., Al-Shamma, O., Fadhel, M., Farhan, L., Zhang, J., & Duan, Y. (2020). Optimizing the Performance of Breast Cancer Classification by Employing the Same Domain Transfer Learning from Hybrid Deep Convolutional Neural Network Model. *Electronics*, vol.9, 3-445.

Atoum, A., Chen, L., Liu, A. X., Hsu, S. D. H., & Liu, X. (2017). Automated online exam proctoring. *IEEE Transactions on Multimedia,* 19(7), 1609-1624.

Brownlee, J. (2020, November 30). Blending Ensemble Machine Learning With Python. *Machine Learning Mastery.* https://machinelearningmastery.com/blending-ensemble-machine-learning-with-python/

Cavalcanti, E. R., Pires, C. E., Cavalcanti, E. P., & Pires, V. F. (2012). Detection and Evaluation of Cheating on College Exams using Supervised Classification. *Informatics in Education,* 11(2), 169–190.

Chaipanha, W., & Kaewwichian, P. (2022). Smote vs. Random Undersampling for Imbalanced Data- Car Ownership Demand Model. *Communications - Scientific letters of the University of Zilina*, 10.26552/com.C.2022.3.D105-D115.

Chan, K., & Stolfo, J. (1997). On the accuracy of meta-learning for scalable data mining. *Journal of Intelligent Information Systems*, 8:1, 5–28.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *JAIR Journal of Artificial Intelligence Research,* vol 16.

Chen, R. C., Dewi, C., Huang, S. W., & Caraka, R. (2020). Selecting critical features for data classification based on machine learning methods. *Journal of Big Data,* 7, 52 (2020).

Chen, Y., Lu, Y., & Moustaki, I. (2020). Detection of two-way outliers in multivariate data and application to cheating detection in educational tests. https://arxiv.org/abs/1911.09408v2.

Chitra, A., & Rajkumar, A. (2015). Plagiarism Detection Using Machine Learning-Based Paraphrase Recognizer. *Journal of Intelligent Systems.*

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

Cohen, A. S., & Wollack, J. A. (2006). Test administration, security, scoring, and reporting. In R. L. Brennan (Ed.) Educational Measurement (4th ed., pp. 355- 386). Westport, CT: American Council on Education/Praeger.

Devries, T., & Taylor, G.W. (2017). Dataset Augmentation in Feature Space. *ArXiv*, abs/1702.05538.

Forman, G., & Scholz, M. (2010). Apples-to-apples in cross-validation studies: Pitfalls in classifier performance measurement. *SIGKDD Explor.*, 12, 49-57.

Geurts, P., Ernst, D. & Wehenkel, L. (2006). Extremely randomized trees. Mach Learn 63, 3–42 (2006). https://doi.org/10.1007/s10994-006-6226-1

Guo, P., Yu, H., & Yao, Q. (2008). The research and application of online examination and monitoring system. *2008 IEEE International Symposium on IT in Medicine and Education,* 497-502.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9), 1263- 1284.

Heaton, J. (2020). An empirical analysis of feature engineering for predictive modeling. arXiv:1701.07852v2. https://arxiv.org/pdf/1701.07852.pdf

Jose, C. (2019, May 12). Anomaly Detection Techniques in Python. *Medium*. https://medium.com/learningdatascience/anomaly-detection-techniques-in-python-50f650c75aaf

Jurich, D. (2011). The Impact of cheating on IRT equating under the non-equivalent anchor test design. Unpublished doctoral dissertation. James Madison University.

Khyani, D., Jakkula, S., Gowda, S., KJ, A., & KR, S. (2021) An interpretation of stacking and blending approach in machine learning. *International Research Journal of Engineering and Technology (IRJET)*, vol. 08, issue 07. https://www.irjet.net/archives/V8/i7/IRJET-V8I7545.pdf

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

Man, K., & Harring, J. R. (2020). Assessing preknowledge cheating via innovative measures: A multiple-group analysis of jointly modeling item responses, response times, and visual fixation counts. *Educational and Psychological Measurement,* 0013164420968630.

McHugh, M. L. (2013). The chi-square test of independence. *Biochemia medica*, *23*(2), 143–149. https://doi.org/10.11613/bm.2013.018

Miao, J. & Niu, L. (2016). A Survey on Feature Selection. *Procedia Computer Science*, 91. 919-926.

Micenková, B., Mcwilliams, B., & Assent, I. (2014). Learning Outlier Ensembles: The Best of Both Worlds – Supervised and Unsupervised. *ODD Workshop on SIGKDD*, pp. 1–4, 2014.

Micenková, B., Mcwilliams, B., & Assent, I. (2015). Learning Representations for Outlier Detection on a Budget. *arXiv:1507.08104*.

Pan, Y., Sinharay, S., Livne, O., & Wollack, J. A. (2022). A machine learning approach for detecting item compromise and preknowledge in computerized adaptive testing. *Psychological Test and Assessment Modeling*.

Pavlyshenko, B. (2018). Using stacking approaches for machine learning models. *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*, pp. 255-258, 10.1109/DSMP.2018.8478522.

Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. *JMLR,* 12, 2825-2830.

Sahu, M. (2016). Plagiarism Detection Using Artificial Intelligence Technique In Multiple Files. *International Journal Of Scientific & Technology Research*, 5(04).

Schaffhauser, D. (2020). Instructors believe students more likely to cheat when class is online. *Campus Technology magazine.*

Schober, P., Boer, C., & Schwarte, L. A. (2018). Correlation Coefficients: Appropriate Use and Interpretation. *Anesthesia & Analgesia*, vol. 126, issue 5, May 2018. doi: 10.1213/ANE.0000000000002864

Schőlkopf, B., Platt, J., Shawe-Taylor, J., Smola, A., & Williamson, R. (1999). Estimating the support of a high-dimensional distribution. *Technical report, Microsoft Research,* MSR-TR-99-87, 1999.

Shu, Z., Henson, R., & Luecht, R. (2013). Using deterministic, gated item response theory model to detect test cheating due to item compromise. *Psychometrika,* 78. 10.1007/s11336-012-9311-3.

Tiong, L., & Lee, H. (2021). E-cheating Prevention Measures: Detection of Cheating at Online Examinations Using Deep Learning Approach -- A Case Study. *arXiv:2101.09841v1.*

Toton, S. L., & Maynes D. D. (2019). Detecting examinees with pre-knowledge in experimental data using conditional scaling of response times. *Frontiers in Education,* (04) June 2019.

Watson, G., & Sottile, J. (2010). Cheating in the digital age: Do students cheat more in online courses. *Online Journal of Distance Learning Administration.*

Zhou, T., & Jiao, H. (2022). Exploration of the Stacking Ensemble machine learning algorithm for cheating detection in large-scale assessment. *Journal of Educational and Psychological Measurement.*

Zopluoglu, C. (2019). Detecting examinees with item preknowledge in large-scale testing using extreme gradient boosting (XGBoost). *Educational and Psychological Measurement,* 79. doi: 10.1177/0013164419839439.