

Image-Based Representation of Item Response Patterns for Test Integrity

Akshay Badola¹ and Mo Zhang²

¹ETS Research Institute, ETS, India

²ETS Research Institute, Educational Testing Services, Princeton 08541, New Jersey, USA

Correspondence: {abadola,mzhang}@ets.org

Abstract

This work re-frames anomalous test response detection in standardized testing as binary image classification by transforming test response logs into fixed-size matrix representations and treating them as grayscale images. Deep learning architectures, including ResNet and Vision Transformer, are evaluated. Experiments demonstrate that the approach is feasible and achieves high recall.

1 Introduction

Ensuring the integrity of tests in large-scale standardized testing is a critical challenge for educational institutions and test administrators, in general. For large scale tests, Multiple-Choice Questions (MCQs) remain the dominant format for large scale assessments due to their scalability and objective grading. However, the integrity of these assessments is frequently compromised with attempts ranging from unauthorized collaboration to the use of external aids. Detecting such violations is crucial for maintaining fairness. Traditional approaches for test integrity (Lee and Haberman, 2013), (Lee and Haberman, 2016), (Richmond and Roehner, 2015) rely on similarity indices between answer sheets, or logistic regression models based on aggregated features (e.g., total time, number of changes). While effective for simple irregularities, these methods struggle with complex, temporal dependencies and high-dimensional interactions present in digital examination logs.

There has been recent interest in using deep neural networks to detect anomalous response patterns in exams (Erdem and Karabatak, 2025), (Kaddoura and Gumaei, 2022). The advantage of using deep networks is that no manual features need to be generated, and

the model can be trained on minimally processed data, which can be beneficial at scale. The deep learning models may also detect non-linear combination of patterns which simpler models or manual features may not capture.

In this work, we propose using deep learning models to re-frame the problem of anomalous test response detection in Multiple-Choice Questions (MCQs) as a binary image classification task. We transform high-dimensional temporal and categorical response logs into a fixed-size matrix representation, effectively treating each test session as a *grayscale image*. Our hypothesis is that as deep learning vision models can capture local 2D patterns in the data, representing the data as images will allow them to similarly find patterns across question items (represented as rows in the matrix) and response features (columns of the matrix). We evaluate three deep learning architectures, two Convolutional Neural Networks (ResNet-20, ResNet-32) (He et al., 2016) and a Vision Transformer (ViT) (Dosovitskiy et al., 2020), on this structured representation and also compare with Gradient Boosted Trees as a baseline.

2 Related Work

The detection of irregular¹ scores in large-scale assessments has evolved from traditional statistical heuristics to sophisticated machine learning and deep learning approaches. Early work in this domain relied on psychometric and statistical methods. Lee and Haberman (Lee and Haberman, 2013) proposed using harmonic regression to monitor scale stability in frequently administered tests, addressing challenges related to seasonal variations and

¹We use the terms irregular, anomalous, fraudulent interchangeably.

unevenly spaced observations. Similarly, Richmond and Roehner (Richmond and Roehner, 2015) examined the effectiveness of analyzing student pair overlaps of correct answers to detect collusion in multiple-choice examinations. More recently, (Lee and Lewis, 2024) introduced a two-stage regression approach to detect section score inconsistencies. They use historical information to identify unusually high scores on easier-to-cheat sections relative to harder-to-cheat ones. Taking an alternative route, (Carrasco et al., 2024) focused on detecting fraudulent communication in online MCQ environments by statistically assessing communication probability based on response concordance and answer time, distinguishing between content production and consumption roles in collusion.

As test process and timing data collection became more ubiquitous, studies like (Lee and Haberman, 2016), (Tang et al., 2019) explored process data in depth and investigated how process and timing data could aid in understanding examinee strategies and demographic relationships to performance. (Kamalov et al., 2021) treated cheating detection as an outlier detection problem, employing recurrent neural networks (RNNs) combined with anomaly detection algorithms to analyze the sequential nature of student assessment data. Kaddoura and Gumaei (Kaddoura and Gumaei, 2022) developed a Convolutional Neural Network (CNN) and Gaussian-based discrete Fourier transform to detect cheating from video frames and speech in real-time, moving beyond hand-crafted features to hierarchical representation learning. Karabatak and Bahaddin (Erdem and Karabatak, 2025) conducted a comprehensive comparison of various machine learning and found that DNNs achieved high accuracy while XGBoost performed best in multi-class scenarios.

While most existing literature focuses on tabular data, sequential logs, or multimedia proctoring, there is a growing interest in novel representation learning techniques. Our work is different in that it recasts the tabular and sequential response logs of MCQs as a binary image classification task. Unlike previous approaches that rely on RNNs for sequence modeling (Kamalov et al., 2021) or autoencoders for latent variable extraction (Tang

et al., 2019), we build upon the well established field of computer vision and utilize computer vision architectures, specifically ResNet and Vision Transformers (ViT), to capture complex, non-linear spatial patterns in the transformed *grayscale* representation of examination sessions. Substantial work exists on modeling tabular data as images, e.g., (Wang and Oates, 2015), (Wang et al., 2016), (Fawaz et al., 2019). Our approach is similar in nature in that we transform MCQ response items as fixed dimensional tabular data and then model them as images. Our approach enables us to outperform a more traditional approach using Gradient Boosted Trees (GBT) in terms of recall on the irregular items.

3 Data and Methodology

For our study, we have used process data comprising 12,336 test administrations. The process data consisted of mouse click data and keyboard presses narrowed to relevant constructs like response input and modification. Each test administration consisted of five distinct sections, each containing 20 multiple-choice or fill-in-the-blank items designed to evaluate numerical and verbal reasoning. Each section was strictly timed, with linguistic domains allotted 30 minutes per section and numerical domains allotted 35 minutes per section, resulting in a total assessment duration of between 160 and 165 minutes. Of the 12,336 tests, 3,443 were flagged as suspicious. The test takers' mouse and keyboard activity was recorded during the test, and the data was processed from those logs.

The following features were used in this study:

1. **start_time**: Start of each visit to the question
2. **dur**: Duration of each visit
3. **n_resp**: Number of responses per visit
4. **n_visit**: Number of visits for each question
5. **seq_sec**: Sequence of the item in the section
6. **sec**: Section type; one of verbal or quantitative.
7. **id**: A unique item identifier. Note that the items' content was not available for this study due to security reasons.

8. **i_type**: The type of the item. One of 10 unique types, e.g., discrete fraction, fill in the blank.
9. **score**: Score obtained for that item.

3.1 MCQ Response Data as Image

We take a direct approach to converting MCQ items to be represented as images. As some of the variables of each MCQ item are discrete while some are continuous, we treat the discrete items as categorical variables and convert them to integers given by their unique index in the set.

Specifically **id**, **i_type**, **seq_sec** are the discrete variables and each is mapped to an integer in the range $\{0, 1, \dots, N_v\}$, where N_v denotes the number of unique categories for variable v .

The variable **start_time** denotes the start of the time for each visit to a particular item in the test. We normalize **start_time** so that the minimum value of **start_time** for each test is 0. Each item can have multiple visits, and therefore, can have multiple values corresponding to **start_time**, **dur** and **n_resp**. As deep learning models require a fixed sized representation, we truncate these variables to have a maximum of n observations per item, with $n \in \{8, 16\}$. We also tried $n = 32$, but the models failed to converge with that.

The rest of the variables are **<n_visit, seq_sec, sec, id, i_type, score>**. Together, the final vector corresponding to each *row* of an item is $\mathbf{x}_i \in \mathbb{R}^{3n+6}$. The final matrix is obtained by concatenating all the rows $\mathbf{X} = [\mathbf{x}_1; \mathbf{x}_2; \dots; \mathbf{x}_{100}] \in \mathbb{R}^{100 \times (3n+6)}$.

$$\underbrace{\text{start_time}}_{\mathbb{R}^n}, \underbrace{\text{dur}}_{\mathbb{R}^n}, \underbrace{\text{n_resp}}_{\mathbb{R}^n}, \underbrace{\text{n_visit, seq_sec, sec, id, i_type, score}}_{\mathbb{R}^6} \equiv \mathbf{x}_i \in \mathbb{R}^{3n+6}$$

An illustration of the aggregation of features into the matrix is given in Figure 1.

3.2 Processing Matrices as Images

Deep learning models require the data to be normalized corresponding to a Gaussian distribution with zero mean and unit variance. Following existing literature (He et al., 2016), we also normalize the data by computing the mean and standard deviation of the data and subtracting and dividing by it, respectively.

We do not however, augment the data with the image processing pipelines as is common

for deep learning models (He et al., 2016), as these are not actual images and do not require rotational or color invariance etc. The normalized data is used to train the models.

3.3 Modeling with CNNs

CNNs use hierarchical *layers* of filter banks $\in \mathbb{R}^{k_1 \times k_2}$, called *kernels*, which process each image in parallel at each layer and then combine the resulting output from the layer. The convolutional *kernels* are the parameters of the model which capture the local spatial relationships in the data.

At layer l of the model consisting of f filter banks, for an input feature map $\mathbf{X}^{(l)} \in \mathbb{R}^{H_l \times W_l}$ and a kernel $\mathbf{k} \in \mathbb{R}^{k_1 \times k_2}$, the output feature map $\mathbf{X}^{(l+1)}$ is computed as:

$$\mathbf{X}_{i,j}^{(l+1)} = \sum_{m=0}^{k_1-1} \sum_{n=0}^{k_2-1} \mathbf{X}_{i+m, j+n}^{(l)} \cdot \mathbf{k}_{m,n}$$

We employ ResNet-20 and ResNet-32 architectures adapted for single-channel input which consists of progressively increasing number of kernels at higher layers along with progressive downsampling of the data. There are also residual operations at every 3 layers which add the data from the previous *block*. ResNet-20 and ResNet-32 are more suited for data of smaller dimensions, motivating our use of these in this study. A key feature of ResNets is the use of batch normalization (Ioffe and Szegedy, 2015) and residual connections.

The ResNet model consists of *layers* and *blocks*, where each layer comprises n blocks. Each *block* consists two sequence of convolution and batch normalization operations in turn.

A residual operation is performed at the first block of each layer. The output is pooled before the final output layer of the model. The reader is referred to (He et al., 2016) for the details of the broader ResNet architecture and its variants. ResNet-20 consists of 3 *layers* of 3 *blocks* each while ResNet-32 consists of 3 *layers* of 5 *blocks* each.

3.4 Modeling with ViT

The Vision Transformer (ViT) model, first introduced in (Dosovitskiy et al., 2020), treats

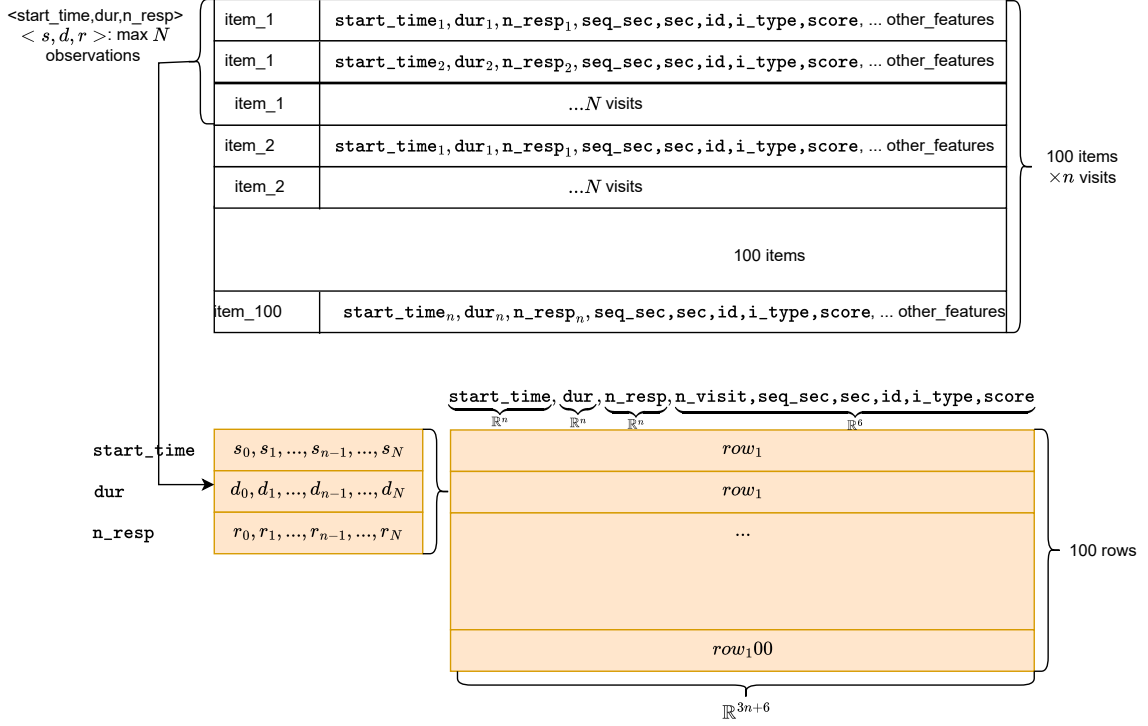


Figure 1: Aggregating process features into (100 times $n+6$) matrix

the input image as a sequence of flattened patches.

For an input image $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ with $C = 1$ for our case, we divide $\mathbf{X}$ into N non-overlapping patches of size $p \times p$. The total number of patches is $N = \frac{H \cdot W}{p^2}$.

Each patch $\mathbf{x}_p \in \mathbb{R}^{p^2}$ is projected via a linear transformation to a D -dimensional embedding $\mathbf{z}_p \in \mathbb{R}^D$: $\mathbf{z}_p = \mathbf{W}_p \mathbf{x}_p + \mathbf{b}_p$ where $\mathbf{W}_p \in \mathbb{R}^{D \times p^2}$ is a learnable projection matrix. The sequence of patch embeddings $\mathbf{Z} = [\mathbf{z}_1; \mathbf{z}_2; \dots; \mathbf{z}_N]$ is the matrix representation input to ViT's transformer blocks.

To preserve spatial information, we add sinusoidal positional encoding, following (Devin et al., 2019). For position $pos \in \{0, \dots, N-1\}$ and dimension $i \in \{0, \dots, D-1\}$, the encoding is defined as:

$$E_{pos, 2i} = \sin\left(\frac{pos}{10000^{2i/D}}\right)$$

$$E_{pos, 2i+1} = \cos\left(\frac{pos}{10000^{2i/D}}\right)$$

The input to the transformer encoder is then $\mathbf{Z}' = \mathbf{Z} + \mathbf{E}_{pos}$. The transformer encoder itself consists of L layers. Each layer applies Multi-Head Attention (MHA) (Vaswani et al., 2017)

and a Feed-Forward Network (FFN). The attention mechanism computes:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right) \mathbf{V}$$

where $\mathbf{Q} = \mathbf{Z}'\mathbf{W}^Q$, $\mathbf{K} = \mathbf{Z}'\mathbf{W}^K$, $\mathbf{V} = \mathbf{Z}'\mathbf{W}^V$ and d_k is the attention dimension.

Multi-head attention generalizes this by projecting $\mathbf{Z}'$ into h different subspaces in parallel. The outputs of all h heads are concatenated along the feature dimension and linearly projected to form the final multi-head attention output:

$$\text{MultiHead}(\mathbf{Z}') = \text{Concat}(\text{head}_1, \dots, \text{head}_h) \mathbf{W}^O$$

where $\mathbf{W}^O \in \mathbb{R}^{D \times D}$ is a learnable output projection matrix. This design allows the model to jointly attend to information from different representation subspaces at different patch positions.

Finally, each layer applies a position-wise Feed-Forward Network (FFN):

$$\text{FFN}(\mathbf{x}) = \text{GELU}(\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1) \mathbf{W}_2 + \mathbf{b}_2,$$

where $\mathbf{W}_1 \in \mathbb{R}^{D \times m}$, $\mathbf{W}_2 \in \mathbb{R}^{m \times D}$, and $\mathbf{b}_1, \mathbf{b}_2$ are learnable parameters.

Layer normalization and residual connections are applied after each sub-layer. The final output is obtained by global average pooling over the sequence dimension and passing through a classification head.

For our experiments we used two configurations of ViT (ViT1, ViT2) both with 9 layers, 8 attention heads per layer and embedding dimension 256. The architectures are designed following existing work (Gani et al., 2022; Lee et al., 2022) for small sized datasets. The difference in the two configurations is patch embedding, wherein the first ViT configuration had a patch embedding of shape $4 \times (3n+6)/9$, while second one had $4 \times (3n+6)$, where $3n+6$ is the number of columns in the MCQ matrix (Section 3.1).

3.5 Gradient Boosted Trees Baseline

We adopt Gradient Boosted Trees (GBT) as a strong tabular baseline for comparison. GBTs are widely recognized for their robustness with heterogeneous feature types and are widely used for modeling various kinds of data.

For GBTs, we follow the following data processing steps:

1. The temporal features are aggregated and converted to five-tuple of (`sum`, `mean`, `max`, `min`, `len`)
2. The rest of the features are converted to one-hot categorical variables

The resulting input for the GBT is a dense tabular vector $\mathbf{x}_{\text{GBT}} \in \mathbb{R}^{d_{\text{GBT}}}$, where d_{GBT} corresponds to the dimensionality after aggregation and encoding.

The GBT model consists of an ensemble of T decision trees constructed sequentially. Each tree f_t is trained to fit the negative gradient of a differentiable loss function (e.g., binary cross-entropy for suspiciousness detection) with respect to the current ensemble predictions. The final prediction for an item is the sum of outputs from all trees:

$$\hat{y} = \sum_{t=1}^T f_t(\mathbf{x}_{\text{GBT}}).$$

For our experiments, we used XGBoost library (Chen and Guestrin, 2016) with 100 trees and with maximum tree depth of 5 and a learning rate of 0.1.

4 Experiments and Results

As mentioned in Section 3.1, we evaluated our models on a dataset of 12,336 samples each consisting of a test administration of 100 questions. Of these, 6,168 were used for training, and rest were set aside as a test set. In the training set, there were a total of 6,168 cases out of which 1,714 were flagged as suspicious and 4,454 were not-flagged. The test set also had 6,168 cases, with 1,729 flagged and 4,439 not-flagged respectively. The data is quite imbalanced with significantly fewer flagged than non-flagged sessions; approximately 25% of sessions are flagged. The results from the experiments are given in Table 1. All the deep learning models were trained with a learning rate of 0.0001 for 20 epochs with AdamW (Loshchilov and Hutter, 2017). The deep learning models were trained with a balanced sampler, so that they were fed roughly equal numbers of valid/invalid cases during training to mitigate the unbalanced nature of the data, as deep learning models can be highly sensitive to the data distribution.

The results provide an interesting trade-off between the precision and recall of the flagged cases. As the data is quite imbalanced (around 25% anomalous scores) we prioritize recall to maximize the detection of actual anomalies, while monitoring precision to minimize false alarms. The Gradient Boosted Tree (GBT) model serves as our strong baseline, achieving a balanced F1-score of 0.759 for the suspicious class. We should note that GBTs often outperform tabular data (Shwartz-Ziv and Armon, 2021), (Grinsztajn et al., 2022). See (Grinsztajn et al., 2022) for a discussion on the issue.

The GBT achieves both high precision and recall. The highest recall, albeit with lower precision, is obtained by ResNet-32 model with $n = 16$. While this does imply a greater number of false positives, it is a useful first layer of screening for anomalous test sessions.

The best overall result except for GBT is obtained by ViT2 configuration with F1-score and accuracy close to that of GBT while having a much higher recall, although with lower precision of 0.625, it is still much higher than other neural models.

Comparing $n = 8$ and $n = 16$ reveals that increasing the context window does not have

Model	Precision	Recall	F1-Score	Accuracy
GBT (Baseline)	0.762	0.757	0.7593	0.8654
Observations $n = 8$				
ViT	0.545	0.917	0.683	0.762
ViT2	0.625	0.890	0.734	0.819
ResNet-20	0.428	0.909	0.582	0.634
ResNet-32	0.587	0.830	0.688	0.789
Observations $n = 16$				
ViT	0.572	0.904	0.701	0.784
ViT2	0.597	0.913	0.722	0.803
ResNet-20	0.575	0.914	0.706	0.787
ResNet-32	0.568	0.927	0.704	0.782

Table 1: Performance of the models for sequence length $n = 8$ and $n = 16$ on the class corresponding to anomalous scores. ViT1, ViT2 refer to ViT models corresponding to the two configurations (Section 3.4). GBT Baseline is presented at the top for comparison. The values in bold indicate the highest value in the column. The values in blue indicate the best result *except* for the baseline GBT.

a significant effect which suggests that more than 8 observations of timing data do not contain enough discriminative signal.

Overall, nearly all the deep learning models achieve a high recall with low precision, again highlighting their propensity for false positives. The role of sequence length for session timings n has no significant effect, but results for $n = 8$ are slightly better than for $n = 16$.

5 Conclusion and Future Work

We have presented a novel approach for anomalous test session detection by treating MCQ process data as images. By transforming temporal and categorical data into a fixed-size matrix, we successfully applied CNN and ViT architectures to the task. Our experiments show that both models achieve higher recall with lower precision than the traditional GBT baseline.

While the image-based pipeline offers a differentiable, end-to-end alternative to manual feature engineering, the lower precision is a limitation. However, for flagging a case as anomalous in the first stage in a multistep review, the low precision, high recall trade-off may be worth having. Future work can explore hybrid representations (e.g., fusing patch embeddings with explicit statistical features), domain-adapted vision encoders, or contrastive pretraining on behavioral traces to bridge this gap.

We also note that as items’ content was not

available to us due to security reasons, it restricts the models’ ability to flag responses based on content. Other datasets which contain item content as well may benefit the deep learning models.

Another limitation and avenue of future work is the number of samples used in the experiments (12,336), as more data can be used to better validate the methods presented.

The class imbalance also presents a challenge, to which the deep learning models are particularly susceptible. Training regimes specifically tailored to alleviate the class imbalance may help the deep learning models achieve parity with respect to precision of GBTs.

References

- M. Carrasco, A. Silva, and Rui Henriques. 2024. Detecting Fraudulent Student Communication in a Multiple Choice Online Test Environment. In *IEEE Trans. on Computational Social Syst.*
- Tianqi Chen and Carlos Guestrin. 2016. Xgboost: a Scalable Tree Boosting System. In *Proc. of the 22nd Acm Sigkdd International Conf. on Knowledge Discovery and Data Mining (kdd)*, pages 785–794. ACM.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre - Training of Deep Bidirectional Transformers for Language Understanding. In *North American Chapter of the Association for Computational Linguistics*.
- A. Dosovitskiy, L. Beyer, Alexander Kolesnikov,

- Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, M. Dehghani, Matthias Minderer, G. Heigold, S. Gelly, Jakob Uszkoreit, and N. Houlsby. 2020. An Image Is Worth 16x16 Words: transformers for Image Recognition at Scale. In *Proc. of the International Conf. on Learn. Represent.*
- Bahaddin Erdem and Murat Karabatak. 2025. Cheating Detection in Online Exams Using Deep Learning and Machine Learning. In *Applied Sciences*.
- Hassan Ismail Fawaz, Benjamin Lucas, G. Forestier, Charlotte Pelletier, Daniel F. Schmidt, J. Weber, Geoffrey I. Webb, L. Idoumghar, Pierre-Alain Muller, and Francois Petitjean. 2019. [Inceptiontime: Finding alexnet for time series classification](#). In *Data mining and knowledge discovery*.
- Hanan Gani, Muzammal Naseer, and Mohammad Yaqub. 2022. [How to train vision transformer on small-scale datasets?](#) In *Proceedings of 33rd British Machine Vision Conference 2022*. BMVA Press.
- Léo Grinsztajn, Edouard Oyallon, and G. Varoquaux. 2022. [Why do tree-based models still outperform deep learning on tabular data?](#) In *Neural Information Processing Systems*.
- Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proc. of the IEEE Conf. on Comput. Vis. and Pattern Recognit.*
- Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proc. of the International Conf. on Mach. Learn.*
- S. Kaddoura and Abdu H. Gumaei. 2022. Towards Effective and Efficient Online Exam Systems Using Deep Learning - Based Cheating Detection Approach. In *Intelligent Syst. with Appl.*
- F. Kamalov, Hana Sulieman, and David Santandreu Calonge. 2021. Machine Learning Based Approach to Exam Cheating Detection. In *Plos One*.
- Seunghoon Lee, Seunghyun Lee, and B. Song. 2022. [Improving vision transformers to learn small-size dataset from scratch](#). *IEEE Access*.
- Yi-Hsuan Lee and S. Haberman. 2013. Harmonic Regression and Scale Stability. In *Psychometrika*.
- Yi-Hsuan Lee and S. Haberman. 2016. Investigating Test - Taking Behaviors Using Timing and Process Data. In *International J. of Testing*.
- Yi-Hsuan Lee and C. Lewis. 2024. A Two - Stage Regression Approach to Detecting Section Score Inconsistency. In *J. of Educational and Behavioral Stat.*
- I. Loshchilov and F. Hutter. 2017. Decoupled Weight Decay Regularization. In *Proc. of the International Conf. on Learn. Represent.*
- Peter Richmond and B. Roehner. 2015. The Detection of Cheating in Multiple Choice Examinations.
- Ravid Shwartz-Ziv and A. Armon. 2021. [Tabular data: Deep learning is not all you need](#). In *Inf. Fusion*.
- Xueying Tang, Zhi Wang, Jingchen Liu, and Z. Ying. 2019. An Exploratory Analysis of the Latent Structure of Process Data via Action Sequence Autoencoder. In *Br. J. of Mathematical & Statistical Psychology*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and I. Polosukhin. 2017. Attention Is All You Need. In *Adv. in Neural Inf. Process. Syst.*
- Zhiguang Wang and T. Oates. 2015. Imaging time-series to improve classification and imputation. In *International Joint Conference on Artificial Intelligence*.
- Zhiguang Wang, Weizhong Yan, and T. Oates. 2016. [Time series classification from scratch with deep neural networks: A strong baseline](#). In *IEEE International Joint Conference on Neural Network*.