A Viable Machine Learning Dataset for Academic Dishonesty Detection in Computer-based Testing

Austin Jerrolds and J.W. Bruce
Department of Electrical and Computer Engineering

1. INTRODUCTION

Testing student knowledge of course material is crucial aspect of learning. The global pandemic forced universities to adopt widespread, computer-based testing environments. Sadly, academic misconduct is at an all-time high. Proctoring computer-based testing proves to be difficult, and often requires methods invasive of student privacy. Currently, most computer-based testing environments collect operational diagnostic data that are indicative of academic misconduct. Using data logging in learning management systems, machine learning methods can be trained to recognize the student actions indicative of dishonest behavior during computer-based testing. Machine learning algorithms require extensive data sets for training. Sizeable data sets of honest and dishonest exam logs are extremely difficult, if not impossible, to obtain. This work describes creation of synthetic data representative of honest and dishonest test taking strategies used by students in a computer-based testing environment. The synthetic data is used to train a decision tree to automatically recognize academic dishonesty during computer-based testing.

2. BACKGROUND

Machine learning training requires vast datasets. Datasets of authentic honest and dishonest student examinations do not exist. Honest students taking an exam will typically move sequentially through the questions, dwell similar durations on each question, and occasionally traverse through the exam again "checking their work". Dishonest students may utilize external resources and finish exams rapidly. Another dishonest approach is students' collaboration manifested by skipping around the exam searching for matching questions. Training and testing dataset should contain both behaviors.

3. OBJECTIVE

- Generate different student profiles for honest and dishonest student types.
- Give each profile their own features that are indicative of their behavior.
- Test the dataset with a decision tree machine learning algorithm.

4. METHOD

The dataset consists of five different student profiles. A profile is randomly chosen with a different weight then a student's exam simulation is conducted resembling that profile. The five student profiles are shown in **Table 1**.

Student Profile	Description
Good Student (5% of dataset)	Uses their time effectively, checks their answers, and scores a high grade.
Average Student (83% of dataset)	May not complete the exam, may or may not check their answers, scores anywhere from a 70 to an 85.
Bad Student (10% of dataset)	Typically, does not finish the exam, scores anywhere from a 40-70
Cheater Fast (1% of dataset)	Finishes the exam in a fraction of the time of everyone else and scores very high.
Cheater Slow (1% of dataset)	Takes the full exam time but spends it jumping around excessively.

Table 1: Student profiles.

Each exam is composed of a feature set that could be collected by learning management systems. The features determine a student behavior. The features examined by this work are listed in **Table 2**.

Feature	Description
Exam Time	Time it takes to complete the exam.
Minus Jumps	Number of times the student jumps backwards.
Forward Jumps	Number of times the student jumps forwards.
Grade	The student's exam score.

Table 2: Generated Feature Set.

Given student profiles and desired feature set, a simulation of a class taking an exam is conducted. **Figure 1** shows a small sample of exam time vs grade.

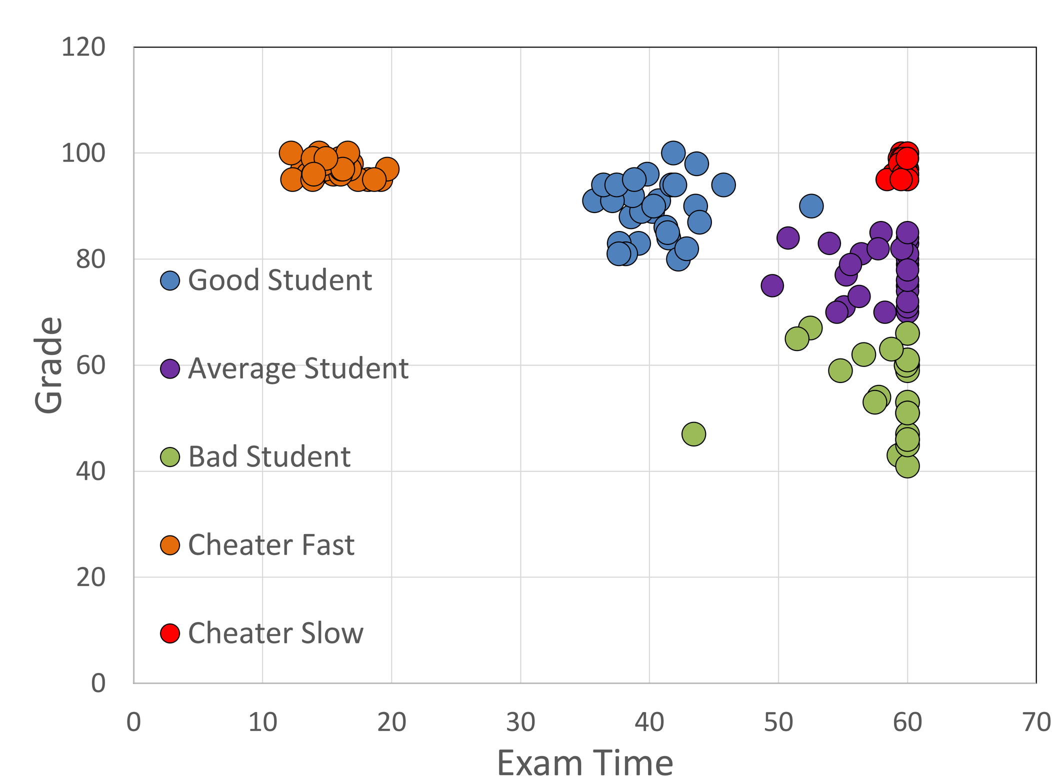

Figure 1: Exam time vs grade for five student populations

When exam time is compared to exam grade, the "Cheater Fast" profile noticeably deviates from the rest of the class. In **Figure 2** when exam grade is compared with "Minus Jumps", the "Cheater Slow" profile is apparent.

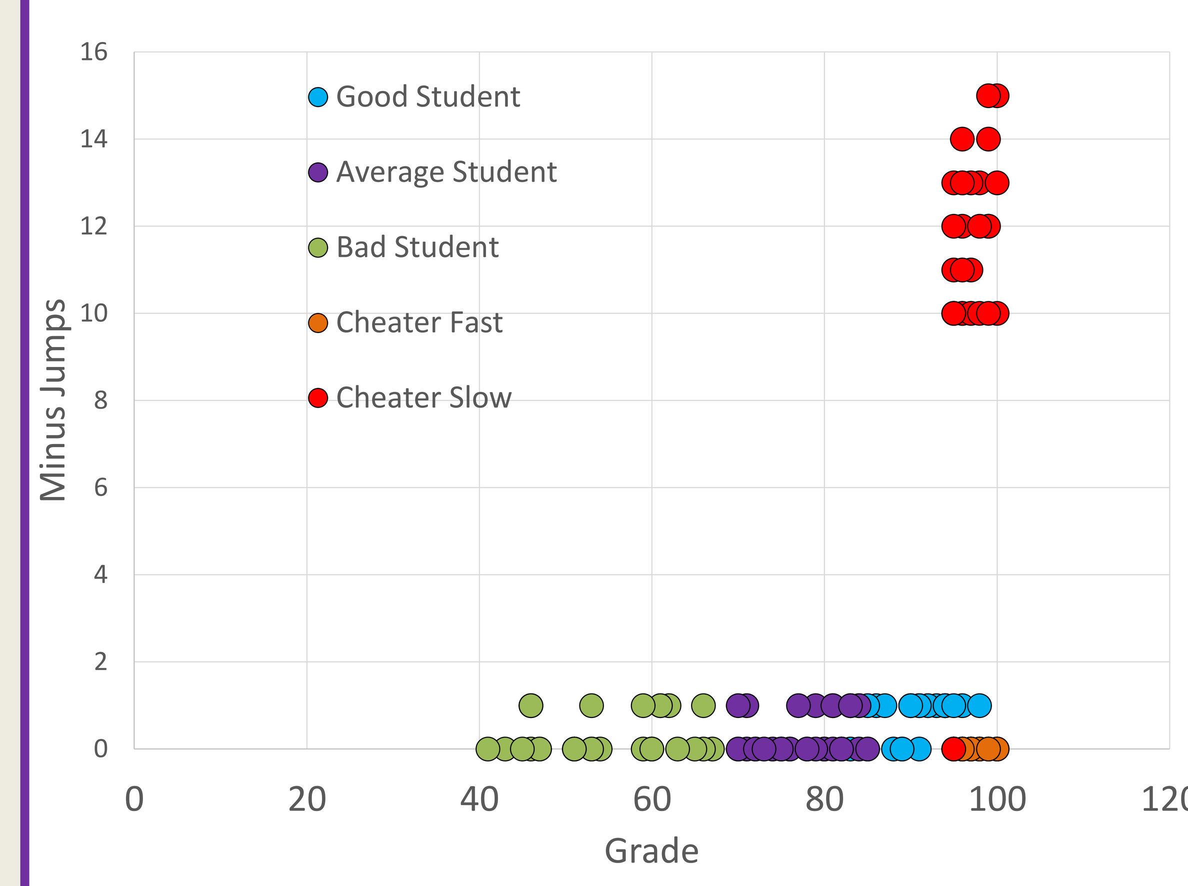

Figure 2: Grade vs reverse jumps for the five populations

With a data set where the features of dishonest students have noticeable differences from honest students, machine learning algorithms can be trained to make determinations more efficiently.

Decision Tree Model

Using a decision tree approach, a machine learning algorithm was trained on 70% of the created synthetic data. After training, inference testing was done with the remaining 30% of data. After the training and testing, performance metrics of accuracy, recall, precision, and F1-score are examined. **Figure 3** displays a visualization of this process.

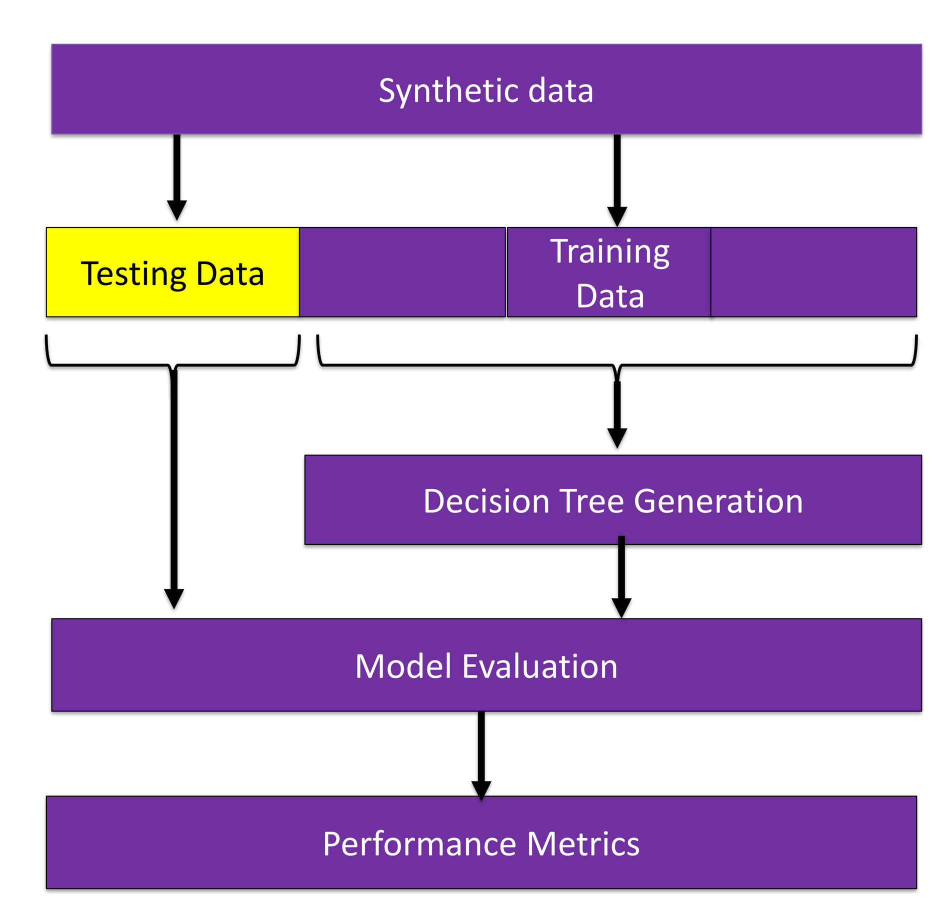

Figure 3: Training, inference, and testing flow

Figure 4 shows a representative visualization of the decision tree algorithm. The decision tree determines the features and threshold most relevant to determining the classification. In this example, "class = 0" denotes an honest student, while "class = 1" is a dishonest student.

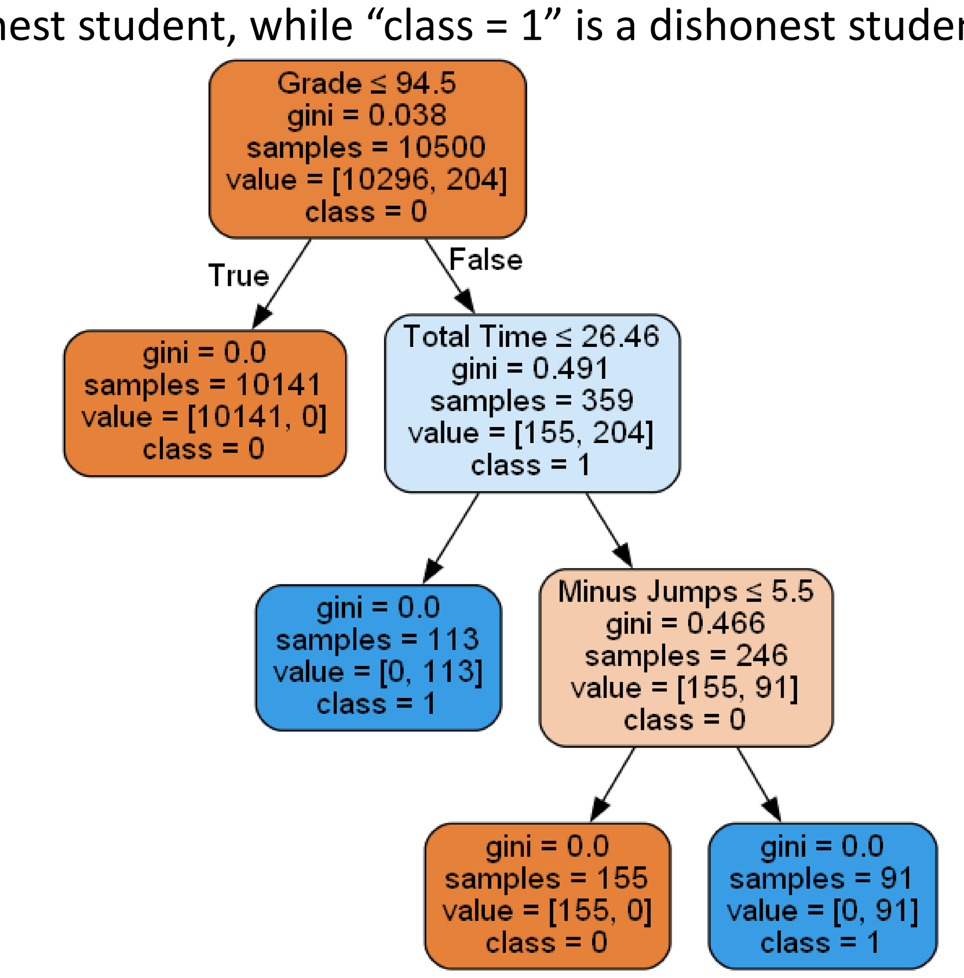

Figure 4: Decision Tree

6. RESULTS

For testing, the decision tree machine learning approach was trained and tested with a synthetic dataset of 15k student exams. After training, inference testing shows: precision = 1.0, recall = 0.99, accuracy = 0.99, and F1-score = 0.995. The results are encouraging and indicate that a robust and sufficiently detailed simulation may generate usable training data.

7. CONCLUSIONS

Although the model possesses high accuracy, training data is synthetic. Statistical parameters governing the creation of synthetic data may not be realistic. However, performance indicates that it is possible to create synthetic data to train machine learning systems to automatically detect academic dishonesty in computer-based testing. Future work will be creation of new features and more realistic feature distributions cause by ambiguity between honest and dishonest behaviors.

8. REFERENCES

[1] M. Ali, "Supervised machine learning," *DataCamp*, 22-Aug-2022. [Online]. Available: https://www.datacamp.com/blog/supervised-machine-learning.
[2] A. Navlani, "Python decision tree classification tutorial: Scikit-Learn Decisiontreeclassifier," *DataCamp*, 28-Dec-2018. [Online]. Available: https://www.datacamp.com/tutorial/decision-tree-classification-python.
[3] P. Patel & J.W. Bruce, "Automatic academic dishonesty detection in computer-based testing," *Proc. Res. & Creative Inq. Day*, Tennessee Tech, pp. 101, April 2022.

Presented at the Tennessee Technological University 18th Research & Creative Inquiry Day, April 19-20, 2023