Towards Automated Machine Learning Detection of Academic Dishonesty in Computer-Based Testing

Parth Patel and J.W. Bruce
Department of Electrical and Computer Engineering

1. INTRODUCTION

The global pandemic forced a rapid shift of learning environments to an online and remote setting. This required examinations to be conducted remotely, however, it is difficult to ensure academic integrity in a remote computer-based testing environment without violating a student's privacy. Without utilizing intrusive technology, the data about the testing process provided from the computer-based testing environment can be collected. Using features extracted from testing logs, a machine learning model could be trained to determine if a student completed the exam honestly or dishonestly. This model can be implemented into an existing computer-based testing environment to allow for suspected academic dishonesty to be automatically flagged for further review.

2. BACKGROUND

There are many ways a student can be dishonest during an online examination. A student utilizing unauthorized resources from the internet, or another student are the most common. When a dishonest student takes an exam, they may move through the exam in random ways visiting problems in unpredictable order. This contrasts with an honest student who normally will answer questions in an exam sequentially. Dishonest students may spend large amounts of time on the same problem without answering while they are searching for or discussing the answer.

3. DATASET

The dataset was created with features extracted from D2L quiz logs from three exams. These logs contain when the quiz was started, completed, movement from page to page, and when a question or page was saved. A total of 19 features were extracted. The dataset consists of 187 total instances, with 8 labeled cheaters and the remaining students labeled as non-cheaters.

4. METHOD

Feature Selection

Due to many features, feature selection was conducted with the WEKA machine learning tool [1] to reduce the number of features. The wrapper subset evaluator was used with the J48 tree classifier. A synthetic dataset generated with SMOTE from the original was used as the input to the evaluator.

The reduced feature set contains 8 features as shown in **Table 1**.

Feature	Description
ExamTime	time to finish exam (in seconds)
MaxPercentTime	max percentage of exam time spent on a page
AvgPercentTime	average percentage of exam time spent on a page
QuestionViewNoEntry	number of pages viewed without changing its question
PlusOneJumps	number of single forward jumps (ie. page 2 to 3)
MinusOneJumps	number of single backward jumps (ie. page 3 to 2)
DirChanges	number of direction changes
NumQuestions	number of questions

Table 1: Reduced feature set.

SMOTE / ADASYN

Since the dataset is highly imbalanced, most learning algorithms will favor the majority class by ignoring the minority class or the inability to discern a pattern in the data. This leads to poor performance on the minority class.

Artificial samples were generated with Synthetic Minority Over-Sampling Technique (SMOTE) [2] and Adaptive Synthetic (ADASYN) sampling [3]. **Figure 1** shows how SMOTE creates samples. ADASYN creates samples similarly to SMOTE, however it prioritizes creating synthetic samples near difficult to classify samples.

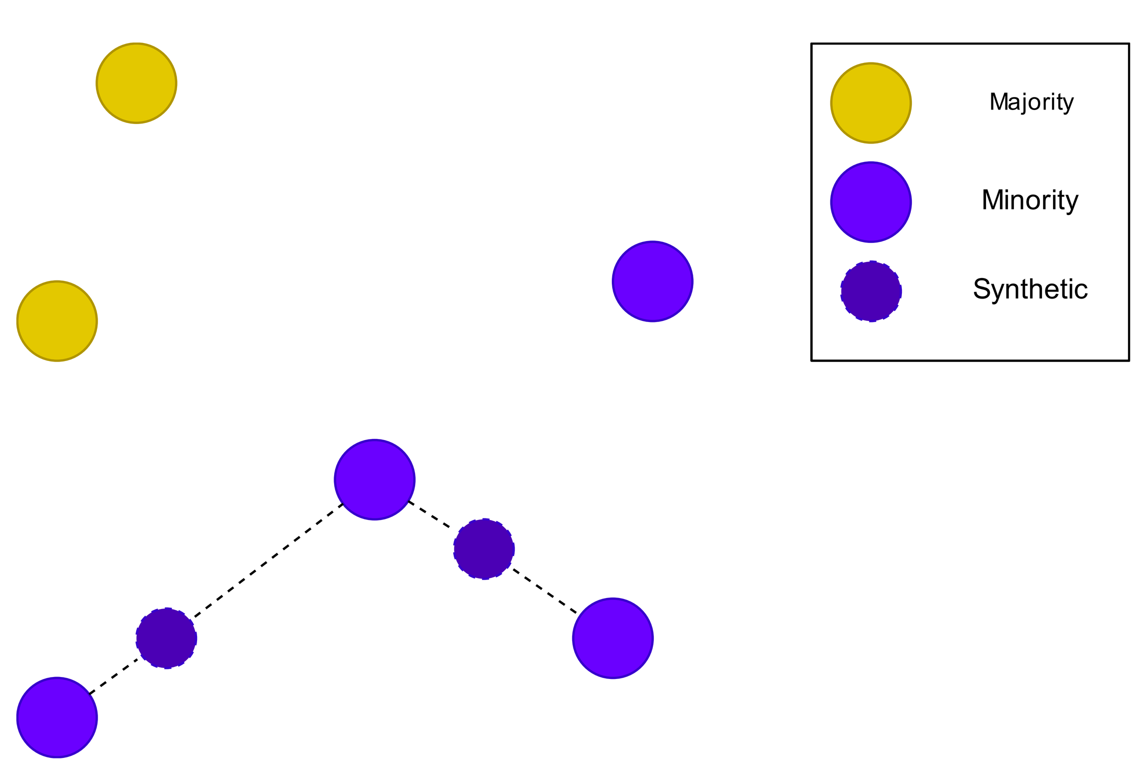

Figure 1: SMOTE algorithm

Model Selection

These supervised and unsupervised machine learning models were chosen because they are commonly used models and provide a wide array of approaches to learning the data.

ML Model
Decision Tree (DT)
Support Vector Machine (SVM)
K-Nearest Neighbors (KN)
Isolation Forest (IF)
One-Class SVM (1SVM)
Local Outlier Factor (LOF)

Table 2: Selected machine learning models.

Model Evaluation

To properly gauge the performances of the models, nested cross validation was utilized. Cross validation (CV) enables a view into how the models would preform if trained on the entire dataset. Which is important for very small datasets due to each sample's importance. However, CV alone can provide biased results[4]. Thus, nested CV is utilized to combat this.

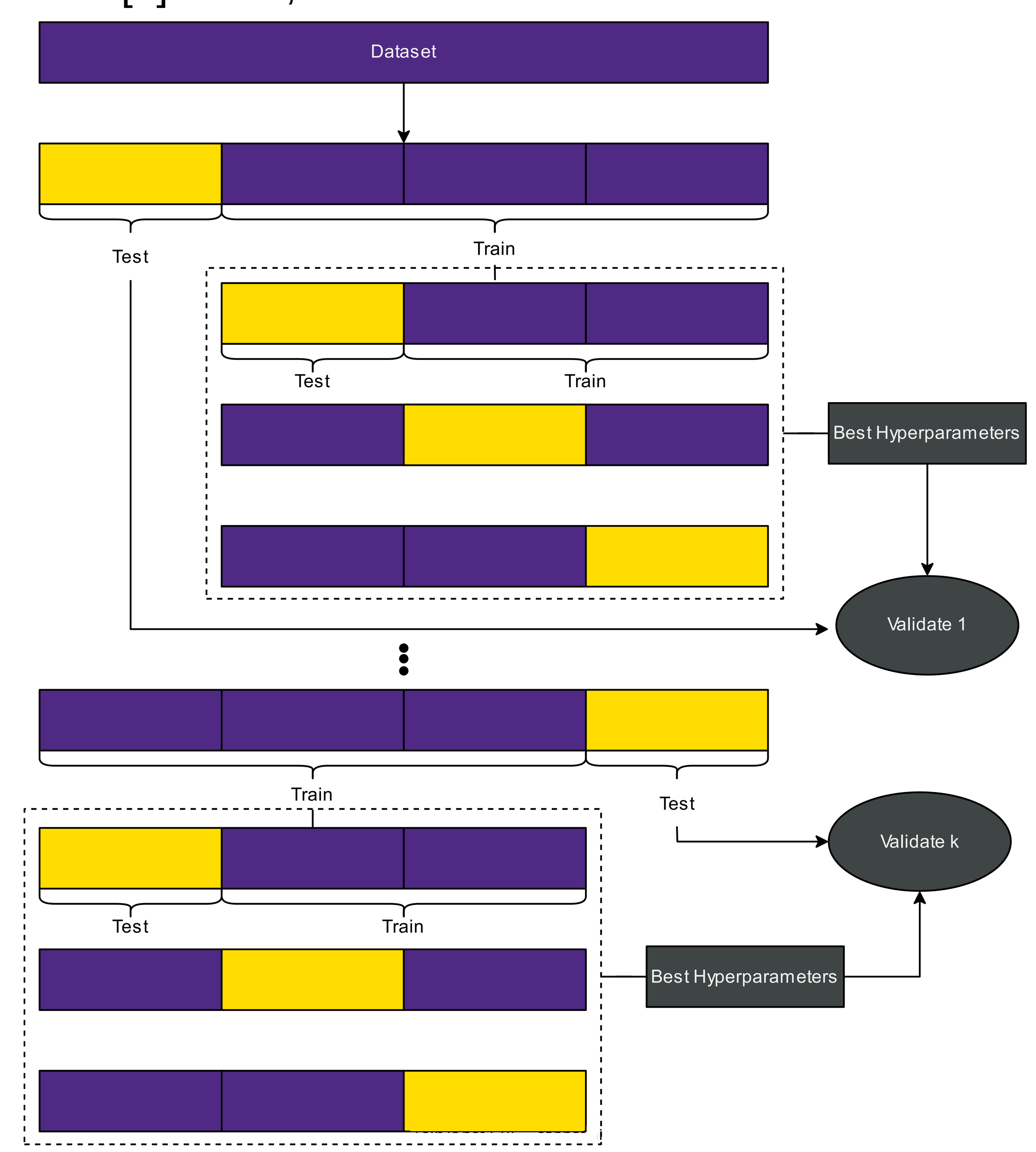

Figure 2: Nested cross validation

6. Results

The metrics of the models after nested CV are in **Figure 3**.

All models performed poorly and with high variance. The high variance is an indication of the models being unstable, slight change in the training dataset causes the model to perform differently.

Overall, all models had low F1 and precision scores. Some models (KN, IF, 1SVM) had high recall scores. However, these models had very low precision scores, indicating these models had high false positive rates.

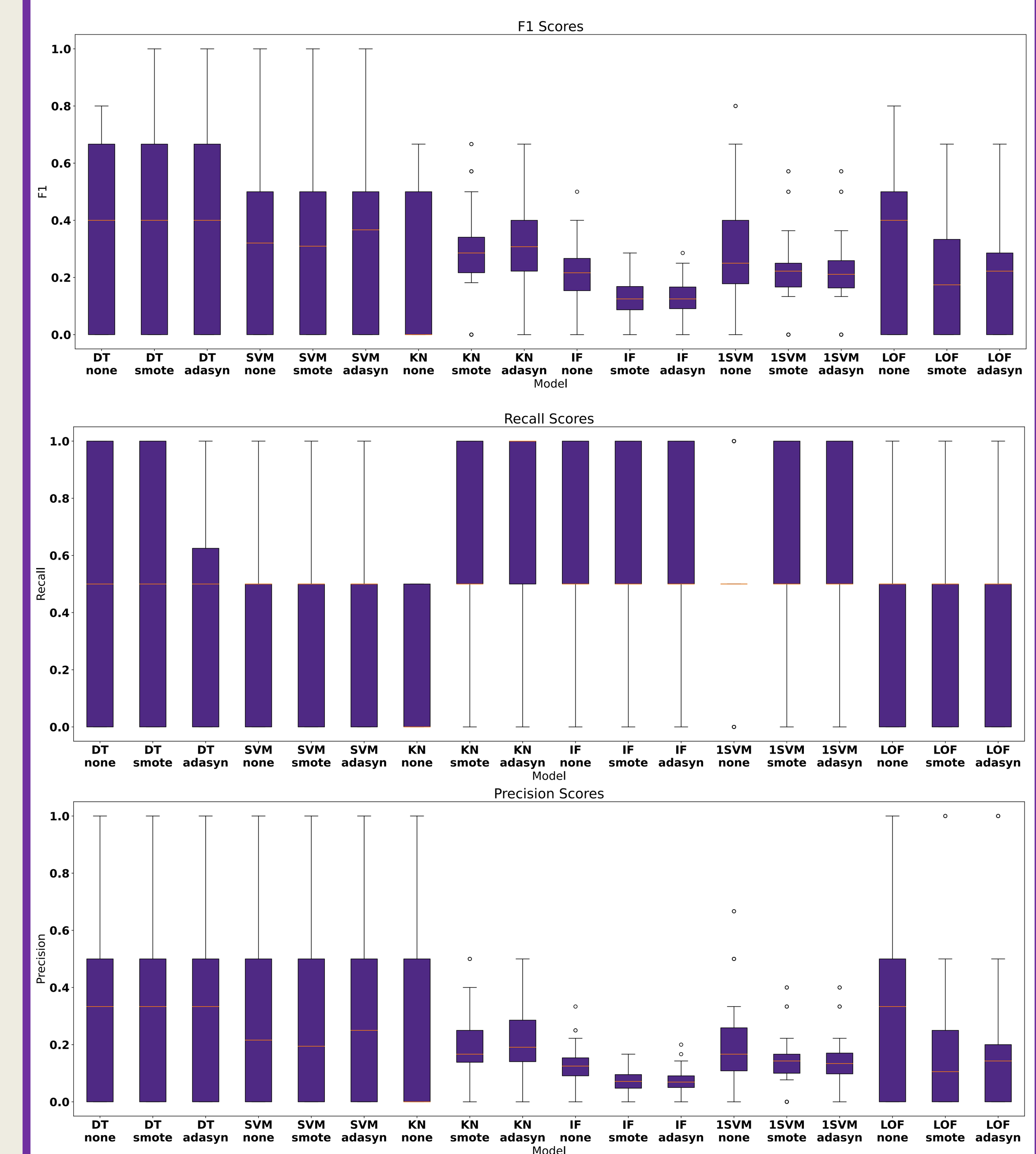

Figure 3: Model metrics after nested CV

6. CONCLUSIONS

This work describes a method for automatic detection of academic dishonesty in computer-based testing. Due to the imbalanced nature of the dataset, synthetic samples were generated utilizing oversampling. Nested cross validation was conducted to evaluate the models on the entire dataset.

The results indicate this method is currently suboptimal at detecting academic dishonesty. However, some models achieved high maximum scores for a few training folds. If more data was available, the high variance in the scores would be reduced while also allowing the oversampling algorithms to utilize more points to synthesize from. However, due to the nature of this data, it is challenging and cumbersome to obtain more data.

7. REFERENCES

[1] Eibe Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.
[2] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, jun 2002
[3] Haibo He, Yang Bai, E. A. Garcia and Shutao Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), 2008, pp. 1322-1328, doi: 10.1109/IJCNN.2008.4633969.
[4] A. Vabalas, E. Gowen, E. Poliakoff, and A. J. Casson, "Machine learning algorithm validation with a limited sample size," PLOS ONE, vol. 14, no. 11, 2019.