Coding Error Detector for Manually-coded Activity Logs: Case Study with Endotracheal Intubation

Sen Yang[1]*, Yichen Zhou[1], Shuhong Chen[1], Alexis Sandler[2], Ivan Marsic[1], Randall S. Burd[2]
[1]Rutgers University, Piscataway, NJ
[2]Children's Nat'l Medical Center, Washington, DC
* Corresponding author email: forrest.yang@rutgers.edu

INTRODUCTION: Some types of medical workflow data can be collected automatically (e.g. EHR data), but others still need to be collected manually (e.g. Surgical Workflow data). This manual coding inevitably introduces human error to the data. In this poster, we use data-driven anomaly detection algorithms to discover human coding errors in workflow data. We implemented the algorithm in a light-weight JAVA application, the Error Detector for Manually-coded Activity Logs. CED-MAL has **three interactive functions (see (A), (B), (C))** to help users locate and interpret these coding errors. We also applied CED-MAL to a preliminary case study on a real endotracheal intubation dataset.

Case Study Dataset:
Our dataset for the endotracheal intubation process (breathing tube insertion) contains 94 patient cases with 1559 total activities. Each activity was manually labeled with one of 15 activity types, a start time, and an end time.

Outlier Detection Algorithms (Figure 1):

- Statistical approach: activities with durations or occurrence times beyond ±2 standard deviations (std) of the average
- Density approach (KNN): activities with times that have k-nearest-neighbor distances beyond ±2 std of the average knn distance
- Clustering approach (K-means): activity in small clusters and at the boundary of large clusters. Num. of cluster is based on silhouette value.

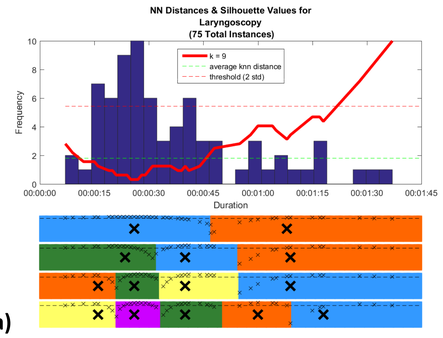
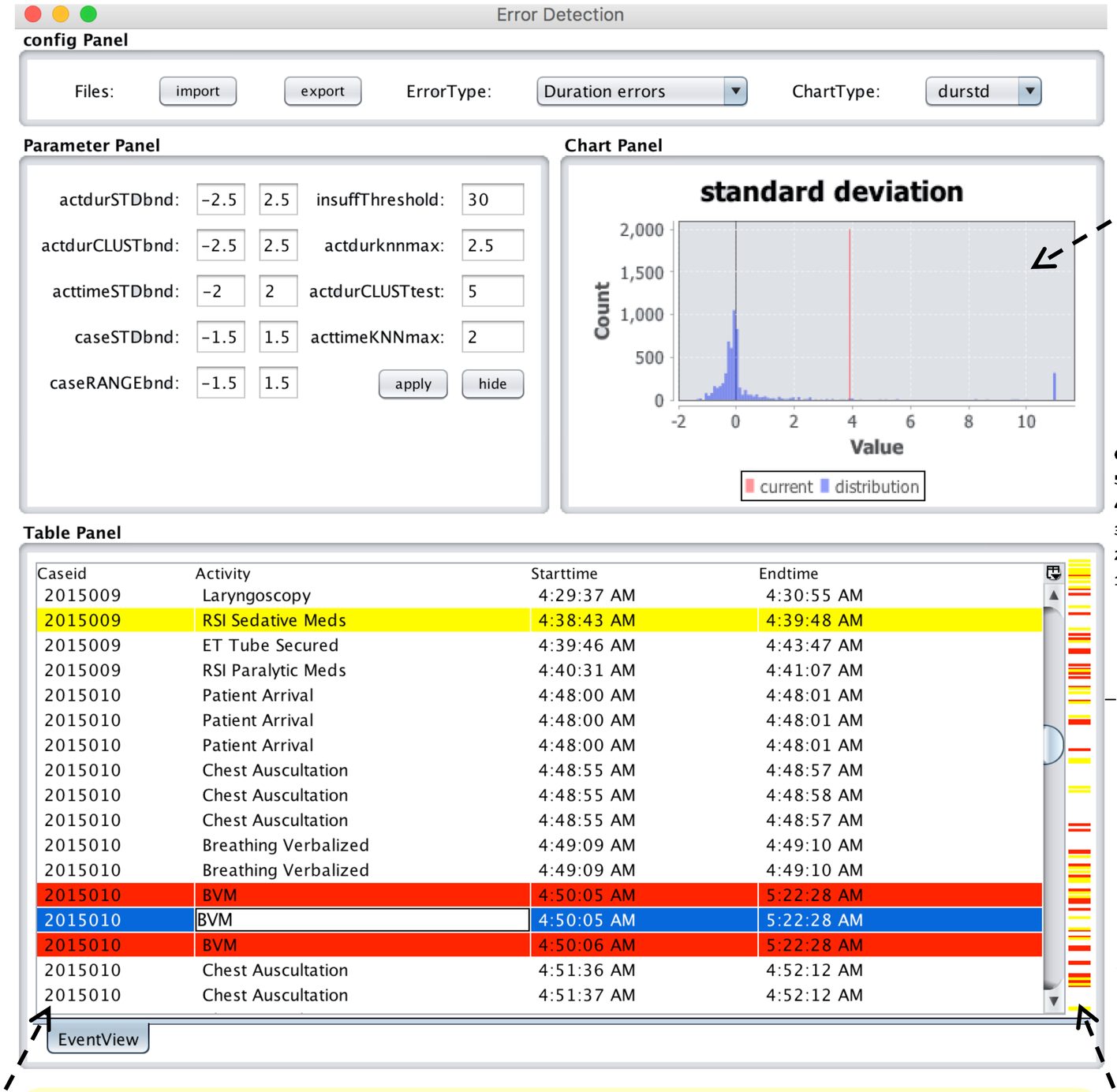


Figure 1. Examples demonstrating the three different anomaly detection algorithms



(A) Data Table View
This shows the imported workflow data. Each row includes an activity's information. The rows of the detected error candidates are highlighted. Users can directly modify the data in this table and output the corrected data with the "export" function. The chart panel above updates to the table's values in real time.

(B) Chart Panel
The chart panel visualizes the error candidate's relationship to the data's distribution. The chart dynamically updates to user behavior (modification, clicking) in the table view. The red line in the chart denotes the location of the data point currently selected in the table view. With this visualization, users can understand why the candidates were labeled as potential errors, as well as how saliently erroneous the candidates are.

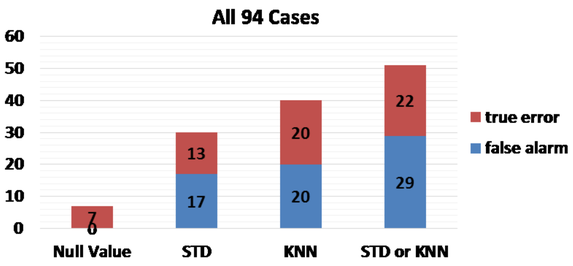


Figure 2. Performance of different algorithms on 94 cases

30 Evaluation Cases

	error	non-error	
detected as error	10	13	23
detected as non-error	5	438	443
	15	451	466

Accuracy	0.96
Precision	0.43
Recall	0.67
F-measure	0.53

Table 1. Performance of (STD or KNN) on 30 Cases

(C) Color-coded Scroll Bar
The color-coded scroll bar gives the user an overall sense of how many error candidates exist in this dataset. Users can easily navigate to the corresponding error candidates by simply clicking on the colored bars. The color (yellow & red) indicates the algorithm's confidence in that error candidate.

Results: Evaluating the detector required finding the actual errors. Because finding the true errors of the entire dataset would be too labor insensitive, we only fully evaluated CED-MAL on 30 cases (Table 1: 0.96 accuracy, 0.43 precision, 0.67 recall, 0.53 F-score). In terms of the entire dataset, we found that CED-MAL achieved 0.43 precision (Figure 2).
Limitations and Future work: Only timestamp coding errors are checked. The dataset may contain other error types (e.g. omitted/repeated encodings, activity name typos, etc.). In future work, we will extend CED-MAL to be a general anomaly detector.
Download: GitHub Link: https://github.com/ShizuoZ/Human-Coding-Checker
Acknowledgement: This research is supported by National Institutes of Health under grant number 1R01LM011834-01A1.