# LESSONS LEARNED FOR AFFECTIVE DATA AND INTELLIGENT TUTORING SYSTEMS

**Keith Brawner**[(a)]

[(a)] United States Army Research Laboratory, Human Research and Engineering Directorate

[(a)]Keith.W.Brawner.Civ@mail.mil

**ABSTRACT**

Human-to-human tutoring is the most effective manner for learning. Human tutors, however, pay attention to the cognitive and affective states of the learners and use the knowledge to adjust instructional strategies. According to theory, data about the learner informs data about instruction informs instructional designs and impacts student learning. The crux of these type of operations is the ability to recognize affect in the learner, but deployed systems to do this have not had significant success with the "one size fits all" type of generalized models. An alternative to this approach is individualized models, which have shown limited success to date in other domains. This paper furthers the investigation of individualized models and validates the approach, showing that it can produce models of acceptable quality, but that doing so does not obviate the experimenter from creating quality generalized models prior to individualizing.

Keywords: Adaptive and Predictive Computer-based Training, Intelligent Tutoring Systems, Architectural Components, Emerging Standards

## 1. INTRODUCTION

Tutoring by an expert human tutor is extraordinarily effective. There is debate among the literature about how effective human tutors are, but it is commonly found to be between one and two standard deviations of improvement; between one and two letter grades (Bloom, 1984; VanLehn, 2011), with the highest reported gains in the range of four standard deviations (Fletcher & Morrison, 2012). Theory indicates that learner data inform learner states which inform instructional strategy selection which influences learning gains (Sottilare, Brawner, Goldberg, & Holden, 2012), shown in Fig. 1. As ITS research moves towards highly adaptable and individualized tutoring, the need to automatically assess the cognitive and affective states of the individual learner for instructional adjustment has been well documented (Department of the Army, 2011; Woolf, 2010), and extensive work has been performed to recognize the emotional state of a learner by incorporating sensors to monitor both behavioral and physiological markers and is discussed in the coming sections (AlZoubi, Calvo, & Stevens, 2009; Calvo & D'Mello, 2010; D'Mello et al., 2005; D'Mello, Taylor, & Graesser, 2007; McQuiggan, Lee, & Lester, 2007).

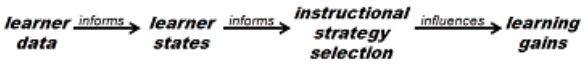


Fig. 1: Learning Effect Chain (Sottilare, 2012)

The traditional manner of incorporating these models is to take measurements of a group of classroom measurement, move the data offline, perform significant amounts of feature extraction, create models, and apply the models to the next set of classroom learners. However, these systems have not been shown to transition well into a field of use due to a variety of reasons including differences in classroom structure, sensor placement, time of day, season of the year,, individual differences, and other reasons (Brawner, 2013). An alternative to this practice are individualized modeling techniques (AlZoubi et al., 2009) using realtime analysis (Brawner, 2013).

However, realtime data presents unique problems for algorithmic modeling purposes. There are four main problems with realtime data, they are 1) the data can be of potentially infinite length, 2) concept detection, 3) concept drift, and 4) concept evolution. The combination of these issues present a problem for whichever type of algorithm is used to solve it. The realtime construction and use approach necessitates a stream model of the data, with the following assumptions, and corresponding design limitations, as (Beringer & Hüllermeier, 2006) outlines:

- Operations must be done on the data as they become available
  - Data ordering is outside of control
  - Data cannot be requested
  - Knowledge about prior points must be encoded, if they are to be related to each other
- The dataset is of infinite length; it is not possible to store or analyze all of the data
  - Data elements are not available for repeated request
- Data must either be saved or discarded

- Practical processing and memory limits necessitate the discard of most data

- There are strict time constraints
  - Data must be processed in real time

- An approximate solution is an acceptable substitute for an ideal one (Considine, Li, Kollios, & Byers, 2004)

These problems are the starting point for an avenue of research into realtime modeling of student state at runtime, rather than *a priori*. However, there are lessons learned from these modeling techniques – the foremost of which is that the models must have feature extraction along the order of the offline models. This paper presents several algorithmic approaches towards dealing with the problem of affective computing at runtime. It discusses lessons learned across 4 algorithmic approaches, 3 labeling approaches, and 5 datasets. The algorithm and labeling approaches are discussed next, followed by a discussion of the various datasets, before performance and lessons learned are addressed.

## 2. ALGORITHMIC APPROACHES TO DEAL WITH REALTIME DATASTREAMS

### 2.1. Introduction

The goal of this research is to further investigate the approach of creating realtime models. It has previously shown promise for small sample sizes (Brawner, 2013; Brawner & Gonzalez, 2016), and the current work is an attempt to expand and validate the approach on a larger and more complete sample. In order to determine if the *approach* is valid, a number of disparate but representative machine learning methods were chosen, including an incremental clustering approach, a neural network approach, a linear regression approach, and a graphical model approach.

Further, it is useful to consider that, since the models are being made for individual learners as they begin, information *from* the learner is available during model construction time. Considering this, it is possible to ask the learner to selectively tag individual states when they are experienced. Each algorithm should take an approach that uses this information to tag the most pervasive (but currently unknown) category. In addition to an outline of each algorithm, modifications to adjust the algorithm for active learning are taken into account.

### 2.2. Clustering Approach

The clustering approach that was taken was an incremental algorithm described in pseudo-code in Fig. 2. The modifications made for semi-supervised and active learning are shown in Fig. 3.

Fig. 2:Pseudo-code description of clustering algorithm

```
For each new point, incrementally
Compare each point to all known centroids
If no cluster is within range of <vigilance>
this point is a new centroid
Else, move the matched cluster <delta> in new point direction
Merge closest centroids based on <vigilance>, if appropriate
Keep track of the number of points in these centroids, and the last
point which modified the centroid, label if possible
```

Fig. 3:Pseudo-code of active learning changes for clustering

```
When a label is requested
Find the largest size centroid which does not currently have a
label, Return the last seen datapoint which modified this centroid
```

### 2.3. Linear Regression Approach

A regression approach was additionally taken based on the superior performance of the logistic regression from the initial models. The Vowpal Wabbit (VW) software package is typically extraordinarily effective for the creation of regression models, and is capable of both one-pass (online) learning and supports the use of active learning by default (Langford, Karampatziakis, Hsu, & Hoffman, 2010). The authors would encourage the readers to see the original publications for a description of the active learning approach used, as their complexity is beyond the scope of this work (Beygelzimer, Hsu, Langford, & Zhang, 2010). A basic description of the algorithm used is described within Fig. 4.

Fig. 4:Pseudo-code of linear regression algorithm; VW

```
Start with for all i: wi= 0          Within the loop:
Get an example:      x ∈ (∞, ∞)
Make a prediction:   y = Σi  wixi
Learn the Truth:     y ∈ [0,1] with importance I
Update the weight:   wi = wi + 2η(y-yi)I
Repeat for specified number of passes or other criteria
```

### 2.4. Neural Network Approach

Adaptive Resonance Theory (ART) is a type of neural network architecture which classifies objects based on the activation of nodes in a structure that is build as new data is presented to it. It was developed to classify data in a one-pass learning environment (Carpenter and Grossberg 1995), and has historic performance roughly equivalent to neural networks, but with significantly reduced training time. In its most basic form, ART draws n-dimensional hypercubes around similar input patterns, where n is the dimension of the input data. Although sometimes viewed as a disadvantage, ART systems are capable of one-pass learning, which makes them appropriate for realtime classification problems. The basic algorithm is described within Fig. 5. Similar alterations to Fig. 3 were added for compatibility with the active learning process.

Fig. 5:Pseudo-code of ART algorithm

```
For each new data point
Compute each neurons' weighted activation to it ( yi = Σwij*xi )
Select the neuron with the highest activation
Test if this neuron in vigilance (xi fuzzyAnd wx < vigilance)
If it is, update the weights:
wi = learningRate*xi + (1-learningRate)*wi
Otherwise, create a new category with xi weights
```

### 2.5. Online Semi-Supervised Growing Neural Gas (OSSGNG)

Neural Gas is a robustly converging alternative to the k-means approach of clustering that finds optimal representations based on feature vectors. These feature vectors construct a topographical map overlaying the data. An example of such an overlay map is included in **Error! Reference source not found.**. This approach has its roots in Self Organizing Maps (SOMs) (Kohonen, 1982) and Neural Gas topologies (Martinetz & Schulten, 1991). Growing Neural Gas (GNG) is an incremental version of Neural Gas which is appropriate for datastream analysis (Holmstrom, 2002), and was initially proposed by Fritzke (Fritzke, 1995). Semi-Supervised GNGs are a further outgrowth of these methods to make use of unlabelled datapoints for classification (Zaki & Yin, 2008). Beyer and Cimiano have modified the initial algorithm to make it appropriate to realtime problems (Beyer & Cimiano, 2011). They present OSSGNG as a topographical mapping algorithm synthesized from the various contributing fields. They examine several metrics for determination of the establishment of clusters, and find that the minimum distance metric has the best performance on problems of interest, which is used in the current work. The algorithm, as well as the realtime modifications and active learning modifications are described in Fig. 6 and Fig. 7.

Fig. 6: Initial Pseudo-code description of OSSGNG algorithm

*Present the set of labeled data (LD) to the network, train only on it, label accordingly*
*Present an input from unlabeled data set (UD), xj,with the previous distance metric*
*Label xj according to the winning node, remove it from UD, enter it into the LD' set*
*Loop until UD set is empty*
*Present LD and LD' to evaluate performance*

Fig. 7: Realtime appropriate pseudo-code description of OSSGNG

*Present a datapoint, finding the two closest items s1 and s2*
*If there is a missing label, assign a label based on the nearest item (unlabeled is possible)*
*Increment ages (detailed originally)*
*Proceed with GNG steps, do not loop to reevaluate*

Active Learning Modifications:
*When a label is requested, find the network of the largest unknown class*
*Compute the centroid of this node-created network*
*Find and request the label of the point closest to the centroid*

### 2.6. Evaluation Approach

Given that this method of model creation is relatively new, it requires a novel method of assessment. To do this assessment, for each individual a model is created over time in each of a supervised, semi-supervised, and unsupervised fashion. At the time of assessment, each of the unlabeled groups is given a label according to the majority-class of the true labels contained within it. As such, an unsupervised approach which draws a single cluster over each datapoint will have a 100% accuracy score, despite a lack of utility. Configuration settings for each algorithm were taken to create a minimal number of categories for labeling.

As a byproduct of the evaluation algorithm, each of the models begins with 100% accuracy, as, in each algorithmic case a single datapoint generates a single cluster and the majority-class of the cluster is correctly labeled. Gradually, as more data about both the user and labels comes available, the overall accuracy of the model decreases. This is seen in the overall trend in each graph. The research question that this paper is trying to answer is whether the approach of creating realtime models on the individual level is useful. As such, it is useful to see the overall effect of the model, and how useful it *would* have been, on average, for a given unit of time. The algorithm which is used to assess the performance of each of the methods is described below in Fig. 6.

Fig. 8:Pseudo-code of assessment algorithm

*For x from 10-100, in increments of 10*
*Feed x% of the data to the algorithm*
*For each class created by unlabeled class boundaries*
*Label this class the majority label of true set*
*Evaluate for AUC ROC accuracy through input of data for classification (next, previous, all seen)*
*Destroy model, loop*

The results presented later were analyzed with the Receiver Operator Characteristic (ROC) benchmark (Hanley, 1989), which plots the proportion of correctly-classified observations from the positive class (true positive rate) against the incorrectly-classified observations (false positive rate). The Area Under the Curve (AUC) of this function was calculated. The AUC ROC is designed to compensate for the misleading figures of "percentage accuracy" for unbalanced data. The AUC ROC measurement allows an algorithm with lower overall error rates, either true positive or false negative, to score well (Hanley & McNeil, 1983), as the all of the categories of possible classification are weighted equally. A measure of 0.5 indicates a simple majority-class classifier. In general, AUC metrics of greater than 0.8 are considered good, while classifiers lower than 0.6 are considered poor; those scoring in the 0.2 range in between those values are considered to be fair. The AUC ROC is comparable to the A' value traditional in machine learning literature.

## 3. DATASETS AND COLLECTION

### 3.1. Dataset #1, #2: Affective and Cognitive Dataset from High- and Low-Cost Sensors

The first two dataset were collected as part of an experiment to evaluate low-cost sensors. College-aged military learners experienced a breadth of learning-relevant emotions while watching videos or playing video games. They were measured by a suite of sensors. Cognitive states, such as distraction, are labeled with a high-cost sensor. Affective states, such as frustration, are labeled with a self-reporting tool called EmoPro (Kokini et al., 2012). There were 14 usable sets of cognitive data,

with 19 usable sets of emotional data. The data was collected while users interacted with a training video game with validated scenarios to induce specific cognitive and emotional states, or while watching videos validated to produce cognitive and emotional states. Models developed under this effort are designed to replace the high-cost sensors measures. This dataset, and the experiment from which it was produced, is referred to as Dataset #1, when using cognitive labels, or as Dataset #2, when using affective labels. The features of this dataset are described in summary in Table 1. More information on the collection, population dynamics, and usability criteria of this dataset can be found in prior work (Brawner, 2013).

### 3.2. Dataset #3: Vigilance and Eyetracking Dataset

The experiment that produced the third dataset was part of a larger suite of experiments, each of which was targeted towards different objectives. The first of these was the objective to examine the relationship of workload and multi-tasking performance as part of a Mixed Initiative Experimental (MIX) testbed, which incorporates theory-driven tasks into a moderately high-fidelity military simulation designed for multi-tasking and physiological data capture (Reinerman-Jones, Barber, Lackey, & Nicholson, 2010). The most relevant experimental purpose is to *create generalized models of physiological response* to situations of changing workload in order to preemptively reduce workload in the future (Barber & Hudson, 2011). This dataset was collected as part of an experiment to evaluate physiological response to situations of changing workload, a cognitively relevant learning state. College students experienced simultaneous tasking on detecting changes and indentifying threats on a displayed monitor. Their cognitive state was monitored by a suite of sensors, with the data cognitively labeled with a high cost sensor. Models developed under this effort are intended to aid in classification of workload, with the intent of having a system compensate during times of high/low operator workload. This dataset, and the experiment that produced it, is referred to as Dataset #3. More information on the college and population dynamics of this dataset can be found in prior work (Brawner, 2013).

### 3.3. Dataset #4, #5: Medical Learning Kinect Dataset, 2013 and 2016

There are two datasets subject to analysis in this paper, one from 2013 and one from 2016 (DeFalco, et al., 2017). They were both collected from a class of United States Military Academy (USMA) cadets as they interacted with a Tactical Combat Casualty Care Simulation (TC3Sim). Participants interacted with the system for approximately an hour of total protocol, while approximately 25 minutes were spent within the simulation. The participants were monitored via within-system interactions as well as via Microsoft Kinect sensor. While the participants interacted with the system, the BROMP protocol (Ocumpaugh, Baker, & Rodrigo, 2012) was used in order to label the data of affective states of the learners as it was observed among the participants in order to label the "ground truth". There are advantages and disadvantages to different labeling schemes (DeFalco, et al., 2017), but in-field observations have been found to be relatively stable over time (Ocumpaugh et al., 2012).

The initial 2013 collection saw the development of various feature extraction methods, which were used in both studies to compare benchmark performance. The features and models from the 2013 study were used in 2016. The raw features of the Kinect data include the center shoulder position, head distance, and top of skull distance. Of the 91 vertices recorded by the Kinect sensor, only three are utilized for posture analysis: top_skull, head, and center_shoulder. These vertices were selected based on prior work investigating postural indicators of emotion with Kinect data (Grafsgaard, Wiggins, Boyer, Wiebe, & Lester, 2014). Derived statistical and windowed features are calculated over top of these items, including the minimum observed, maximum observed, median, variance; each of these features is additionally calculated for 5/10/20 second windows. Further and more specific information on the dataset can be found in other works (DeFalco, et al., 2017; Rowe, Mott, & Lester, 2014; Rowe, Lobene, & Sabourin, 2013).

### 3.4. Summary of Dataset Features

Due to paper length limits, a discussion of all of the features, meaning behind the features, and derived features of all of the sensors, across each of the five datasets, is not possible. Feature extraction was performed on dataset #1 and #2, with simple statistical measures (mean over a time period, difference from last observation, etc.). No effort for feature engineering was attempted on dataset #3. Extensive feature engineering was performed on dataset #4 and #5, and more information can be found in the publication which discusses the process (Paquette et al., 2016). The sensors, labels, and measurement across the number of datasets are presented in Table 2. Dataset #5 did not have any occurences of Boredom during observation, so models of Boredom are not considered in any of the models created.

Table 1 – Summary of sensors and ground truth across datasets (***bold italics*** indicates ground truth, * indicates feature extraction previously performed)

| Dataset | Sensor | Measure |
|---|---|---|
| ***#1**** | ***ABM EEG*** | ***HighEngagement***<br>***Distraction***<br>***Workload*** |
| ***#2**** | ***EmoPro*** | ***Anger***<br>***Boredom***<br>***Fear*** |
| #12 | Neurosky EEG | Alpha1, Alpha2, Gamma1, Gamma2, Delta, Beta1, Beta2, Theta, Attention, Meditation |

| | | |
|---|---|---|
| #12 | Zephyr HxM | Heart |
| #12 | Motion (custom) | Motion |
| #12 | Chair (custom) | Chair1-8 |
| #12 | Eye Tracking (custom) | LeftEyePupilDiameter |
| ***#3*** | ***FaceLab 5*** | ***IndexofCognitveActivity*** |
| #3 | FaceLab 5 | FixationDuration, PupilDiameter |
| ***#45**** | ***BROMP*** | ***Boredom, Confusion, Engaged Concentration, Frustration, Surprise*** |
| #45 | Microsoft® Kinect | Various, see text. |

## 4. RESULTS

Given that a single user had ten datapoints, one for each of ten percentage points among the timeline, a summary presentation of data is required. Firstly, it is useful to note that performance starts high, as a single point represented in a single cluster has a 100% accurate classifier to categorize it.

Models with a performance benchmark below 0.6 are typically considered unusable. The above table shows that the best performance models using realtime methods for Dataset #1 and Dataset #3 are below any reasonable margin of usability. Among the usable models, the most common usable techniques consistently are the neural and grouping techniques represented by ART and K-means clustering.

Summary results can be presented in multiple manners – graphically, numerically in a number of single-model tables, or via summary table. The results are presented in this manner in Fig. 9, Fig. 10, and Table 2, respectively. Many results from many models and many studies are presented in this paper, with the goal of answering the "does this approach work?" question. Table 2 is best to answer this question, but it is based on the data shown in Fig. 9, Fig. 10, and many other similar figures, a substantial fraction of which have been published elsewhere (Brawner, Gonzalez, 2013).

Fig. 9: Unsupervised, Supervised, and Semi-Supervised performance of the ART algorithm over time, Dataset #5

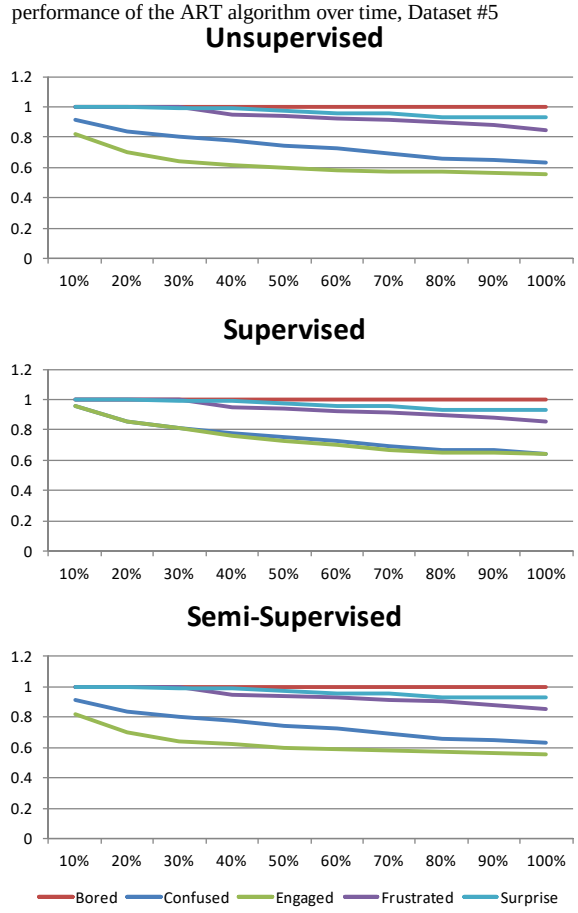


Fig. 10: Boredom model qualities with supervised ART algorithm, Dataset #2

| User | 20% | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% | Avg |
|---|---|---|---|---|---|---|---|---|---|---|
| 4134 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | **1.000** |
| 4133 | 0.96 | 0.92 | 0.92 | 0.77 | 0.70 | 0.71 | 0.68 | 0.72 | 0.58 | **0.773** |
| 4131 | 0.97 | 0.66 | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 | **0.939** |
| 4127 | 0.63 | 0.67 | 0.60 | 0.60 | 0.53 | 0.51 | 0.62 | 0.51 | 0.65 | **0.591** |
| 4121 | 0.80 | 0.95 | 0.82 | 0.81 | 0.83 | 0.83 | 0.81 | 0.84 | 0.77 | **0.829** |
| 4111 | 1.00 | 1.00 | 1.00 | 0.75 | 0.73 | 0.79 | 0.83 | 0.74 | 0.79 | **0.846** |
| 4115 | 1.00 | 1.00 | 1.00 | 0.95 | 0.63 | 0.91 | 0.52 | 0.79 | 0.58 | **0.821** |
| 4135 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | **1.000** |
| 4136 | 1.00 | 1.00 | 1.00 | 0.60 | 0.60 | 0.60 | 0.60 | 0.60 | 0.60 | **0.733** |
| 4137 | 0.91 | 0.78 | 0.74 | 0.73 | 0.76 | 0.78 | 0.80 | 0.81 | 0.82 | **0.792** |
| 4101 | 0.66 | 0.66 | 0.66 | 0.64 | 0.65 | 0.74 | 0.72 | 0.64 | 0.63 | **0.665** |
| 4117 | 1.00 | 0.67 | 0.52 | 0.67 | 0.72 | 0.51 | 0.56 | 0.58 | 0.59 | **0.648** |
| 4102 | 0.85 | 0.79 | 0.78 | 0.85 | 0.80 | 0.67 | 0.71 | 0.76 | 0.81 | **0.780** |
| 4105 | 0.80 | 0.84 | 0.84 | 0.69 | 0.62 | 0.66 | 0.63 | 0.62 | 0.66 | **0.708** |
| 4104 | 1.00 | 1.00 | 0.75 | 0.87 | 0.80 | 0.74 | 0.66 | 0.73 | 0.75 | **0.810** |
| 4107 | 1.00 | 1.00 | 1.00 | 0.75 | 0.79 | 0.74 | 0.79 | 0.79 | 0.79 | **0.848** |
| 4106 | 0.65 | 0.75 | 0.67 | 0.75 | 0.70 | 0.64 | 0.70 | 0.70 | 0.72 | **0.699** |
| 4112 | 0.98 | 0.88 | 0.88 | 0.88 | 0.78 | 0.78 | 0.65 | 0.60 | 0.58 | **0.779** |
| 4132 | 1.00 | 1.00 | 0.79 | 0.89 | 0.84 | 0.85 | 0.87 | 0.86 | 0.75 | **0.873** |
| Average | **0.906** | **0.872** | **0.839** | **0.800** | **0.760** | **0.760** | **0.743** | **0.750** | **0.739** | **0.796** |
| Total Usable (avg ROC >0.6): | | | | 18 | | Percent Usable: | | | 95% | |

## 5. DISCUSSION

Methods for realtime model creation are fundamentally handicapped when compared to traditional model creation methods. They are tasked to predict with a *fragment* of the total data, under *strict time constraints*. Given how unlikely the offline-created models are to transfer to a field of use, a practitioner should consider a small drop in overall accuracy to be acceptable. Large drops in accuracy, or models which poorly model the

Table 2 – Summary of Model Performances. TIE indicates similar performance for ART and clustering unless otherwise noted. Instances where models are usable are bolded, usable instances (semi-supervised only) are italicized.

| Dataset | Best Supervised Model / Quality | Best Semi-Supervised Model / Quality | Best Un-Supervised Model / Quality | Prior Benchmark |
|---|---|---|---|---|
| #1 (Distract) | TIE .552<br>Inck & GNG | GNG .538 | Inck .537 | .81 |
| #1 (Engage) | TIE .560 | TIE .555 | TIE .555 | .80 |
| #1 (Workload) | Inck .529 | VW .520 | GNG .524 | .82 |
| #2 (Anger) | **ART .776** | ***Inck .677*** | **TIE .652** | <0.6 |
| #2 (Boredom) | **ART .796** | ***Inck .626*** | **TIE .612** | .79 |
| #2 (Fear) | **ART .841** | ***Inck .810*** | **TIE .805** | .83 |
| #3 (ICA) | GNG .505 | GNG .506 | GNG .504 | N/A |
| #4 (Boredom) | **Inck .886** | ***Inck .888*** | **Inck .891** | 0.528 |
| #4 (Confusion) | **Inck .820** | ***Inck .820*** | **Inck .831** | 0.535 |
| #4 (E. Conc.) | **Inck .949** | ***Inck .765*** | **Inck .952** | 0.532 |
| #4 (Frust.) | **ART .939** | ***TIE .939*** | **ART .941** | 0.518 |
| #4 (Surprise) | **ART .954** | ***ART .954*** | **ART .955** | 0.493 |
| #5 (Confusion) | **ART .630** | ***Inck .640*** | **Inck .750** | 0.489 |
| #5 (E. Conc.) | **Inck .595** | ***Inck .595*** | **Inck .647** | 0.546 |
| #5 (Frust.) | **TIE .851** | ***ART .851*** | **TIE .851** | 0.331 |
| #5 (Surprise) | **TIE .932** | ***TIE .932*** | **TIE .932** | 0.51 |

underlying phenomenon should, of course, not be considered for application.

These methods failed on two of the above datasets (#1 and #3), experienced only moderate success on one of the above datasets (#2), and experienced significant success on two of the datasets (#4, #5). In a discussion, we should consider the things that both failure and success have in common. In short, these are the frequency of data, quality of labels, and quality of feature extraction.

### 5.1. Quality of Labels

Firstly, there should be a note about the variance in the quality of the labels. There are a variety of labeling techniques, each with its own advantages and disadvantages (Brawner & Boyce, 2017). Dataset #1 and #3 used an automated sensor system to assign label individual cognitive states. These cognitive state labels change multiple times per second. In contrast, Dataset #2 used self-report labels to label a state experienced during the total, but short, interaction. This creates a more stable label, but may not be an accurate representation of ground truth due to the time between labeling and the experience. In contrast, The BROMP protocol labels of Dataset #4 and #5 are relatively stable at approximately 1 minute resolution, but taken during live interactions without delay. The best overall performance was observed with the stable labeling techniques that are likely to be observed in real-world performance.

The datasets with the worst performance (#1, #3) used labeling techniques for *cognitive* states, while the datasets with the best performance (#2, #4, #5) used labeling techniques for *affective* states. D'Mello would categorize affect as either states, traits, moods, or emotions (D'Mello, Blanchard, Baker, Ocumpaugh, & Brawner, 2014), which are defined partly by the longevity of their experience. The "sweet spot" for ITS is in relatively short affective emotions (anger, fear) or affective-cognitive blends (confusion, concentration), and *not* long-lasting moods (depression) or ephemeral cognitive states (distraction). The methods described in this work function well inside the "sweet spot", but generally fail outside of it. Simply, the methods presented here work well for the types of occasions where it would be used – relatively short term classifications of relatively short term states.

### 5.2. Frequency of Data

The frequencies of the Datasets varied significantly, as shown in Table 3. Each of the method selected is capable of realtime performance, but unusually high frequency is one possible reason for poor model performance with Dataset #3, and may have served to compound errors with labeling quality.

Table 3 – Dataset Frequencies

| Dataset | Frequency |
|---|---|
| #12 | ~1 Hz |
| #3 | ~3500 Hz |
| #45 | ~0.7 Hz |

### 5.3. Quality of Feature Extraction

The best performing datasets (#4, #5) already had significant feature extraction performed in addition to useful-quality models based on the extracted features. Moderately performing datasets (#1, #2) had only statistical features (mean, difference, etc.). Poorly performing dataset (#3) had no manufactured features whatsoever. Generally speaking, the higher quality, as shown by their utility in offline-created models, of the features indicates that the realtime models will perform better. It is worthwhile to note that effort was spent in order to make an apples-to-apples comparison of model

– the realtime and offline models were always made with the exact same features. A data scientist making models for utility is not bound by these constraints and is free to manufacture features as they see fit.

## 6. CONCLUSIONS

Generally speaking, the technique of realtime modeling, as opposed to offline modeling, *works*. It should not be forgotten that the alternative to this type of approach is the traditional process involving online collection, offline creation, and online usage, which has yet to see results in the field of practice; it *doesn't work*. With that said, however, there are places where this approach shines and places where the approach fails.

The first area where this approach fails is when the learner state is highly fluid in comparison to the collection window. If the learner state changes at 30 Hz, but data is collected 1 Hz, it is difficult to algorithmically determine learner state. The experimenter should be aware of instances where the learner state is fluid. The second major area of failure is when proper feature extraction hasn't been performed. Feature extraction is a useful process to offline signals and online signals alike; failure to isolate proper features will produce unusable models.

Naturally, the opposite of the above failures indicates the successes – realtime models can be created if the state is changing slowly relative to the feature collection window and when proper features have been considered prior to interpretation by models. The goal of this paper is to set up provide a guide to creating realtime models and to set expectations for performance. The best realtime models observed thus far have had the following features:

- They attempt to model an affective-cognitive combined state, which is relatively stable on the order of minutes.
- They make use of offline experiments for validation of feature extraction techniques

An example experimental setup for affective state detection within an intelligent tutoring systems may have the following features:

- Hardware sensors of physiological state - ideally standoff sensors such as the Microsoft Band for GSR and heartrate measures
- Existing feature extraction shown useful in other contexts – such as a 300ms sliding window of signal power on a GSR signal
- Participant able to label affect states as they come available; a system able to request these items
- Use of one of more machine learning measures, such as ART or incremental clustering.

The intelligent tutoring system described above must also be able to perform the other functions of an intelligent tutoring system, such as changing the instruction to better suit the learner. Modeling the state of the learner is only the start of the process of customizing instruction and feedback towards individual needs.


## ACKNOWLEDGMENTS

The author would like to thank Benjamin Goldberg for assistance in collection/provision of the initial dataset and numerous enlightening conversations, Lauren Reinerman and Julian Abich for the collection/provision of the second dataset, and Ryan Baker, Jonathan Rowe, and Jeanine DeFalco for aiding in the collection of the final two datasets. The research described herein has been sponsored by the U.S. Army Research Laboratory. The statements and opinions expressed in this article do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.


## REFERENCES


AlZoubi, O., Calvo, R., & Stevens, R. (2009). Classification of EEG for Affect Recognition: An Adaptive Approach. *AI 2009: Advances in Artificial Intelligence*, 52-61.

Barber, D., & Hudson, I. (2011). Distributed logging and synchronization of physiological and performance measures to support adaptive automation strategies. *Foundations of Augmented Cognition. Directing the Future of Adaptive Systems*, 559-566.

Beringer, J., & Hüllermeier, E. (2006). Online clustering of parallel data streams. *Data Knowl. Eng, 58*(2), 180-204.

Beyer, O., & Cimiano, P. (2011). *Online semi-supervised growing neural gas.* Paper presented at the Workshop New Challenges in Neural Computation 2011.

Beygelzimer, A., Hsu, D., Langford, J., & Zhang, T. (2010). Agnostic active learning without constraints. *arXiv preprint arXiv:1006.2588*.

Bloom, B. S. (1984). The 2-Sigma Problem: The search for methods of group instruction as effective as one-to-one tutoring,. *Educational Researcher, 13*(6), 4-16.

Brawner, K., & Boyce, M. (2017). *Establishing ground truth on pyschophysiological models for training machine learning algorithms: Options for ground truth proxies*. Paper presented at the International Conference on Augmented Cognition a part of the Human Computer and Intelligent Interaction (HCII) multi-conference.

Brawner, K. W. (2013). *Modeling Learner Mood In Realtime Through Biosensors For Intelligent Tutoring Improvements.* Doctor of Philosophy in EECS, University of Central Florida.

Brawner, K. W., & Gonzalez, A. J. (2016). Modelling a learner's affective state in real time to improve intelligent tutoring effectiveness. *Theoretical Issues in Ergonomics Science, 17*(2), 183-210.

Calvo, R. A., & D'Mello, S. (2010). Affect detection: An interdisciplinary review of models, methods, and their applications. *Affective Computing, IEEE Transactions on, 1*(1), 18-37.

Considine, J., Li, F., Kollios, G., & Byers, J. (2004). *Approximate aggregation techniques for sensor*

*databases.* Paper presented at the 20th IEEE International Conference on Data Engineering.

DeFalco, J., Rowe, J., Paquette, L., Georgoulas-Sherry, V., Brawner, K.W., Mott, B., Baker, R.S., Lester, J. C., (2017), Detecting and Addressing Frustration in a Serious Game for Military Training. International Journal of Artificial Intelligence and Education (IJAIED). Springer.

D'Mello, S., Blanchard, N., Baker, R., Ocumpaugh, J., & Brawner, K. (2014). I Feel Your Pain: A selective Review of Affect-Sensitive Instructional Strategies. In B. Sottilare, A. Graesser, X. Hu & B. Goldberg (Eds.), *Design Recommendations for Intelligent Tutoring Systems: Volume 2 - Instructional Management* (Vol. 2, pp. 35): Army Research Laboratory.

D'Mello, S. K., Craig, S. D., Gholson, B., Franklin, S., Picard, R. W., & Graesser, A. C. (2005). Integrating Affect Sensors in an Intelligent Tutoring System *Affective Interactions: The Computer in the Affective Loop Workshop at 2005 International Conference on Intelligent User Interfaces* (pp. 7-13). New York: AMC Press.

D'Mello, S. K., Taylor, R., & Graesser, A. C. (2007). Monitoring Affective Trajectories during Complex Learning. In D. S. McNamara & J. G. Trafton (Eds.), *Proceedings of the 29th Annual Cognitive Science Society* (pp. 203-208). Austin, TX: Cognitive Science Society.

Department of the Army. (2011). The U.S. Army Learning Concept for 2015. TRADOC.

Fletcher, J., & Morrison, J. (2012). DARPA digital Tutor: Assessment data. *Retrieved from Institute for Defense Analyses: http://www. acuitus. com/web/pdf/D4686-DF. pdf*.

Fritzke, B. (1995). A growing neural gas network learns topologies. *Advances in neural information processing systems, 7*, 625-632.

Grafsgaard, J., Wiggins, J., Boyer, K. E., Wiebe, E., & Lester, J. (2014). *Predicting learning and affect from multimodal data streams in task-oriented tutorial dialogue.* Paper presented at the Educational Data Mining 2014.

Hanley, J. A. (1989). Receiver operating characteristic (ROC) methodology: the state of the art. *Critical reviews in diagnostic imaging, 29*(3), 307.

Hanley, J. A., & McNeil, B. J. (1983). A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology, 148*(3), 839-843.

Holmstrom, J. (2002). *Growing neural gas.* Master's Thesis, Uppsala University.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological cybernetics, 43*(1), 59-69.

Kokini, C., Carroll, M., Ramirez-Padron, R., Hale, K., Sottilare, R., & Goldberg, B. (2012). *Quantification of trainee affective and cognitive state in real-time.* Paper presented at the The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC).

Langford, J., Karampatziakis, N., Hsu, D., & Hoffman, M. (2010). Vowpal Wabbit 5.0 *Neural Information Processing Systems 2010 Conference Tutorial*.

Martinetz, T., & Schulten, K. (1991). *A" neural-gas" network learns topologies*: University of Illinois at Urbana-Champaign.

McQuiggan, S., Lee, S., & Lester, J. (2007). Early prediction of student frustration. *Affective Computing and Intelligent Interaction*, 698-709.

Ocumpaugh, J., Baker, R. S. J. d., & Rodrigo, M. M. T. (2012). Baker-Rodrigo Observation Method Protocol (BROMP) 1.0. Training Manual version 1.0. *New York, NY: EdLab.*(Technical Report).

Paquette, L., Rowe, J., Baker, R., Mott, B., Lester, J., DeFalco, J., . . . Georgoulas, V. (2016). Sensor-Free or Sensor-Full: A Comparison of Data Modalities in Multi-Channel Affect Detection. *International Educational Data Mining Society*.

Reinerman-Jones, L., Barber, D., Lackey, S. J., & Nicholson, D. (2010, July 17-20). *Developing methods for utilizing physiological measures.* Paper presented at the Applied Human Factors and Ergonomics Society Conference 2010.

Rowe, J., Lobene, E. V., & Sabourin, J. (2013). *Run-Time Affect Modeling in a Serious Game with the Generalized Intelligent Framework for Tutoring.* Paper presented at the AIED 2013 Workshops Proceedings Volume 7.

Rowe, J. P., Mott, B. W., & Lester, J. C. (2014). *It's All About the Process: Building Sensor-Driven Emotion Detectors with GIFT.* Paper presented at the GIFTSym2, Pittsburgh, PA.

Sottilare, R. A. (2012, September 2012). *Considerations in the Development of an Ontology for a Generalized Intelligent Framework for Tutoring.* Paper presented at the International Defense & Homeland Security Simulation Workshop Vienna, Austria.

Sottilare, R. A., Brawner, K. W., Goldberg, B. S., & Holden, H. A. (2012). The Generalized Intelligent Framework for Tutoring (GIFT).

VanLehn, K. (2011). The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. *Educational Psychologist, 46*(4), 197-221.

Woolf, B. P. (2010). A Roadmap for Education Technology: National Science Foundation.

Zaki, S. M., & Yin, H. (2008). A semi-supervised learning algorithm for growing neural gas in face recognition. *Journal of Mathematical Modelling and Algorithms, 7*(4), 425-435.

**Keith Brawner, Ph.D.** is a researcher within the Learning in Intelligent Tutoring Environments (LITE) Lab within the U. S. Army Research Laboratory's Human Research & Engineering Directorate (ARL-HRED) in Orlando, Florida. He has 11 years of experience within U.S. Army and Navy acquisition, development, and research agencies. He holds a Masters and PhD degree in Computer Engineering with a focus on Intelligent Systems and Machine Learning from the University of Central Florida. His current research is in machine learning, active and semi-supervised learning, ITS architectures and cognitive architectures. He manages research in adaptive training, semi/fully automated user tools for adaptive training content, and architectural programs towards next-generation training.