Trust Prediction Modeling in Human-Autonomy Teaming Applications

Joe Hesse-Withbroe
Univ. Colorado Boulder
Boulder, CO, USA

Sarah Leary
Univ. of Colorado Boulder
Boulder, Colorado

Pravesh Vadapalli
Univ. of Colorado Boulder
Boulder, Colorado

***Abstract*—Human-autonomy teaming is becoming increasingly common in military applications, space applications, and in daily life. Trust must be established and appropriately calibrated within a team to perform effectively. Surveys are considered the gold standard in trust measurement, but are operationally obtrusive and do not allow for continuous monitoring. Biosignal data, collected via physiological sensors, potentially offer an unobtrusive and continuous trust inference method. This work aims to build trust prediction models using both a Neural Network and maximum likelihood Bayesian parameter learning for a Hidden Markov Model. Both personalized (n=1) and cohort (n=4) models are trained. Furthermore, some models are trained using physiological data that would only be captured in a laboratory-based environment as they are more obtrusive to work (deemed the ”Full” models), while other models are trained using physiological data that could be captured via wearable sensors (”Simple” models). Overall, the ”Simple” personalized models trained by the Neural Network appear to achieve the best performance overall. We suspect the ”Full” model network is training on too many features and thus is finding underlying relationships that are unrepresentative of the actual physiological mechanism behind trust. For future modeling with neural networks, we recommend to reduce the feature space further than was performed in this work.**

***Index Terms*—Human-autonomy teaming, trust in automation, human trust prediction**

I. Introduction

Autonomous systems are increasingly employed in military applications, space missions, and daily life [4], [7], [16]. Humans often act as a ”supervisor” of these autonomous systems, thus forming a Human-Autonomy Team (HAT). To have a high-performing and effective team, trust must be established and appropriately calibrated within the team [9]. Failure to appropriately calibrate trust could lead to misuse of the system, if the human over-trusts a poorly performing system, or disuse of the system, if the human under-trusts a well-performing system [14]. To understand how trust manifests and changes in these HATs, surveys and other self-report methods are employed. This method is considered the ”gold standard” of trust measurement, as they are empirically-derived and validated [10], [19]. However, these methods are obtrusive in that they force a human to stop work to indicate their trust, so they cannot be employed in an operational environment. Furthermore, these methods can only capture a static measure of trust, but trust is actually a dynamic and continuously-changing construct [3], [18]. Thus, an alternative inference method is needed in order to understand trust and constantly monitor trust in operational environments. Biosignal data from physiological sensors, such as cerebral blood flow from functional Near Infrared Spectroscopy (fNIRS), skin conductance responses from Electrodermal Activity (EDA) sensors, and gaze data from eye-tracking glasses, have shown promise in inferring trust dynamics [5], [6], [15], [20]. Thus, collecting biosignal data in an operational setting could allow for *unobtrusive* and *continuous* trust prediction in *real-time*.

The long-term goal of HAT-related research is to create a closed-loop control system, as is shown in Fig. 1 [1]. The human’s trust (inferred unobtrusively from biosignals) is intended to be ”actuated” or ”damped” by actions of the system via the ”Trust Management Algorithm” and the ”Human-Machine Interface”. These elements could include the reliability or the explainability of the system itself. The damping allows the human to calibrate their trust appropriately and minimize the error between the ”actual trust level” and the ”desired trust level”. The research presented in this paper focuses specifically on the ”Psychophysiological Trust Sensing”, as is circled by the dotted line.

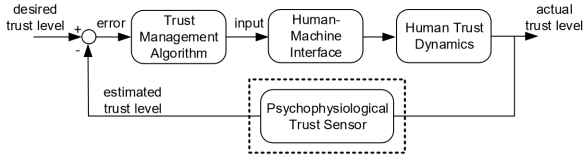

Fig. 1: A block diagram of a feedback control system for achieving trust management in HAT applications. The scope of this work includes psychophysiological trust sensor modeling. Adapted from [1].

Most studies to date have focused on finding physiological correlates with trust [5], [6], [15], [20]. In other words, previous work has aimed to identify the underlying physiological mechanisms behind trust. While this is useful from a perspective of understanding how trust manifests physiologically in a human (i.e., the fundamental science), further work is required to make this operationally useful.

Some researchers have extended past this work and aimed to build trust prediction models. Akash et al. (2018) recruited participants to interact with an autonomous vehicle, while simultaneously collecting brain electrical activity data from electroencephalogram (EEG) and skin conductance responses from EDA. Throughout the task, participants were asked to report whether they ”Trusted” or ”Distrusted” the autonomous vehicle. Trust prediction models were later developed by training personalized (n=1 participant) and cohort (n=48 participants) quadratic discriminant classifier models. This group was able to achieve a mean prediction accuracy of 78.55% and 71.22% for their personalized models and cohort models, respectively.

Similarly, Yi et al (2023) recruited 45 participants to interact with an autonomous vehicle, while simultaneously collecting heart electrical activity data from electrocardiogram (ECG), skin conductance responses from EDA, gaze data, and task-based measures. Throughout the task, participants were administered a single-item questionnaire that asked ”On a scale from 0 to 100, how much do you trust the system?” Overall, 16 trust reports were collected for each participant. After data collection, this group separated trust reports into a binary ”high trust” and ”low trust” group to simplify model predictions. This group then separated the data into a training (80%) and test (20%) set and trained the parameters of a neural network with an ADAM optimizer. The neural network had a convolutional neural network layer, a long short-term memory network layer, and a fully connected layer. The neural network achieved an accuracy score of 73.7% across all 45 retained participants.

Lastly, Zhao et al. (2023) employed a Hidden Markov Model (HMM) to predict latent states in HAT applications. In this example, participants played a search-and-rescue game with an autonomous system teammate. The researchers collected behavioral data on the participants while performing the task as well as a ground truth collective intelligence score - these data served as the ”observations” and ”latent states”, respectively. They then were able to train transition and emission probability distributions which helped to decode unseen behaviors and output a predicted collective intelligence score.

The research presented in this paper aims to combine many approaches seen in previous work. First, the study aims to employ an HMM with biosignal data as the ”observable data” and trust as the ”latent state”. In parallel, this work also aims to build a neural network that is trained on the biosignal data as well as the ground truth trust data collected from our experiment. Furthermore, this work plans to compare the prediction capabilities of personalized (n=1) and cohort (n=4) models. Lastly, this works aims to investigate differences in performance between a large set of physiological sensors that would unlikely be employed in an operational environment due to their long setup time and potential discomfort (e.g., fNIRS) against a subset of physiological sensors that can be incorporated into wearable devices (e.g., EDA).

II. Background

A. Hidden Markov Models

1) Markov Chains: A Markov chain is a model that gives us the likelihoods of sequences that random variables, states, take. A fundamental assumption of a Markov Chain is that the future states in the sequence only depend on the current state and that the past states are not correlated when making this prediction.

Markov Assumption: $P(s_i = a|s_1 \ldots s_{i-1}) = P(s_i = a|s_{i-1})$ where $s_1, s_2..s_i$ is a sequence of the states.

Formally a Markov Chain is a tuple (S, T, π)

- States (S): A bounded set of states of size N.
- Transition Probability Matrix (T): A matrix of size NxN with elements a_{ij} that represent the probabilities of transitioning from state i to state j such that $\sum_{j=1}^{n} a_{ij} = 1 \quad \forall i$.
- Initial State Distribution (π): A probability distribution over the initial states, indicating the probability of starting the chain in each state.

A Markov chain is implemented when one needs to compute the probability of a sequence of observable states. A Hidden Markov Model is used when these states are unobservable.

2) Hidden Markov Models: Hidden Markov Models (HMMs) are based on the augmentation of the Markov Chain. HMMs have two underlying assumptions.

HMM assumption 1: $P(s_i = a|s_1 \ldots s_{i-1}) = P(s_i = a|s_{i-1})$
similar to that of a Markov model, such an HMM is called a first-order HMM.

HMM assumption 2: $P(o_i|s_1 \ldots s_i, \ldots, s_T, o_1, \ldots, o_i, \ldots, o_T) = P(o_i|s_i)$
output observation o_i is dependent only on the state s_i that produced it and not on any other state or observation.

Formally, HMM can be represented as a tuple (S, O, T, E, π)

- States (S): A finite set of hidden states.
- Observations (O): A finite set of observations $o_1, o_2 ... o_T$: a sequence of T observations, each one drawn from the vocabulary V.
- Transition Probability Matrix (T): Probability matrix for transitions between hidden states.
- Emission Probability Matrix (E): a sequence of observation likelihoods or emission probabilities each representing a probability of an observation o_t(from the vocabulary set V) being generated from state s_i.
- Initial State Distribution (π): A probability distribution over the initial hidden states.

3) Viterbi Algorithm: Given a set of observations, the task of finding the most likely sequence of hidden states that generate the observations is called decoding. One of the most common algorithms used for decoding in HMMs is the Viterbi algorithm.

Two matrices of size $T \times |\mathcal{S}|$ are constructed:

$P_{t,s}$ contains the maximum probability of ending up at state s at observation t, out of all possible sequences of states leading up to it.

$Q_{t,s}$ tracks the previous state that was used before s in this maximum probability state sequence.

Let π_s and $a_{r,s}$ be the initial and transition probabilities respectively, and let $b_{s,o}$ be the probability of observing o at state s. Then the values of P are given by the recurrence relation:

$$P_{t,s} = \begin{cases} \pi_s \cdot b_{s,o_t} & \text{if } t = 0, \\ \max_{r \in \mathcal{S}} \left(P_{t-1,r} \cdot a_{r,s} \cdot b_{s,o_t} \right) & \text{if } t > 0. \end{cases} \tag{1}$$

The formula for $Q_{t,s}$ is identical, except that $\max$ is replaced with $\arg\max$. The Viterbi path can be found by selecting the maximum of P at the final timestep and following Q in reverse. Please refer Appendix for the pseudocode.

4) Baum Welch Algorithm: Baum Welch algorithm is an Expectation Maximization(EM) algorithm that finds the maximum likelihood estimates of HMM parameters given a set of observations.

We can describe a hidden Markov chain by $\theta = (T, O, \pi)$. The Baum–Welch algorithm finds a local maximum for $\theta^* = \arg\max_\theta P(O \mid \theta)$ (i.e., the HMM parameters θ that maximize the probability of the observation). Initialize $\theta = (A, B, \pi)$ with random initial conditions, or alternatively, set them using prior information about the parameters if available. This can speed up the algorithm and guide it towards the desired local maximum.

The Baum Welch algorithm can be divided into forward, backward and update steps which will be explained in the below subsubsections.

Forward Step

The forward procedure calculates $\alpha_i(t) = P(o_1, o_2 \ldots, o_t, s_i \mid \theta)$, the probability of observing the sequence $o_1, o_2, \ldots, o_t$ and being in state i at time t. This is computed recursively:

$$\alpha_i(1) = \pi_i b_i(o_1), \tag{2}$$

$$\alpha_i(t+1) = b_i(o_{t+1}) \sum_{j=1}^{N} \alpha_j(t) a_{ji}. \tag{3}$$

To prevent numerical underflow for longer sequences, scaling is applied to α in the forward and β in the backward procedure below.

Backward Step

Let $\beta_i(t) = P(o_{t+1}, \ldots, o_T \mid s_t, \theta)$ be the probability of the ending partial sequence $o_{t+1}, \ldots, o_T$ given starting state i at time t. We calculate $\beta_i(t)$ as follows:

$$\beta_i(T) = 1, \tag{4}$$

$$\beta_i(t) = \sum_{j=1}^{N} \beta_j(t+1) a_{ij} b_j(o_{t+1}). \tag{5}$$

Update Step

We can now calculate the temporary variables using Bayes' theorem:

1. $\gamma_i(t)$ represents the probability of being in state i at time t given the observations O and the HMM parameters θ:

$$\gamma_i(t) = P(S_t = i \mid O, \theta) = \frac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^{N} \alpha_j(t)\beta_j(t)}, \tag{6}$$

2. $\xi_{ij}(t)$ represents the probability of being in states i and j at times t and $t+1$, respectively, given the observation sequence O and the HMM parameters θ:

$$\begin{aligned} \xi_{ij}(t) &= P(S_t = i, S_{t+1} = j \mid O, \theta) \\ &= \frac{\alpha_i(t) a_{ij} \beta_j(t+1) b_j(o_{t+1})}{\sum_{k=1}^{N} \sum_{w=1}^{N} \alpha_k(t) a_{kw} \beta_w(t+1) b_w(o_{t+1})} \end{aligned} \tag{7}$$

The denominators of $\gamma_i(t)$ and $\xi_{ij}(t)$ are the same; they represent the probability of observing O given the parameters θ. The Hidden Markov model parameters θ can now be updated

$$\pi_i^* = \gamma_i(1), \tag{8}$$

$$a_{ij}^* = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)}, \tag{9}$$

$$b_i^*(v_k) = \frac{\sum_{t=1}^{T} 1_{o_t = v_k} \gamma_i(t)}{\sum_{t=1}^{T} \gamma_i(t)}, \tag{10}$$

These steps are iterated until convergence. The given algorithm is for a single sequence of observed data but this can also be easily translated into multiple sequences. It is important to note that it is possible to overfit. Baum welch also doesn't guarantee global maximum which is why it is advised to initialize θ.

Method to initialise θ

The state transition matrix is initialized by counting the number of transitions from a given state to the next state and then normalizing this matrix. A similar method is used for the emission matrix.

III. Experiment

A. Protocol

The data used in these models were collected during a laboratory based experiment where a participant teams with a simulated autonomous system in a computer screen based task. The participant and simulated autonomous system have a shared goal of identifying ground troop movement in pieces of data gathered from ground imaging satellites. Every 45 seconds during the task, the participant is asked to rate their trust in their autonomous system teammate via a Visual Analog Scale in the form of a slider. These trust reports are considered the true trust state of the participant and are compared against the trust values outputted from the predictive models. Each participant reports their trust 8 times per trial. There are 6 trials per session and there are 4 sessions total (over the course of four days). Thus, each participant reports their trust a total of 192 times. Prior to the start of each trial, there is a 45 second baseline period where the participant is instructed to stare at a cross hair in the center of the screen, but not think about the task. The participant interacts with a new autonomous system during each testing session. Each autonomous system varies in its reliability and explainability, in an effort to produce a broad range of trust reports. For more information about the task itself, see Sung et al. (2024).

While the participant is performing the task, they are equipped with numerous physiological sensors, including electrocardiogram (ECG) in a three-lead configuration across the chest, a respiration chest strap, EDA electrodes on the middle and pointer fingers, and a 20 channel fNIRS system (8 sources, 7 detectors). These data serve as predictors in the trust prediction models.

B. Data Pre-Processing Methods

Data pre-processing is performed in MATLAB. The ECG data is band pass filtered between 1 Hz and 100 Hz to remove baseline drift and filter out electromyographic noise. R-DECO is used to extract R-peaks [11]. The EDA data is smoothed with a third-order Savitzky-Golay filter and then Continuous Decomposition Analysis is performed in LedaLab [2]. The respiration data was band pass filtered between 0.05 Hz and 3 Hz with a second order Butterworth filter. Each channel of fNIRS data is bandpass filtered between 0.016 Hz and 0.5 Hz.

Then, all cleaned time-series data is divided into 45 second epochs (i.e., the time in between trust reports) and all features are extracted for each epoch. All features then have the pre-trial baseline period subtracted out to account for inter-trial and inter-session differences. For each trust report, we have 197 features (4 ECG, 9 EDA, 4 respiration, 180 fNIRS) that can be used as predictors. In other words, for 192 trust reports, we have 192 sets of features (consisting of 197 features each).

Due to the high dimensionality of fNIRS data (i.e., high number of predictors per trust report), it is advantageous to employ a feature reduction technique to mitigate the risk of model over fitting. The Least Absolute Shrinkage and Selection Operator (LASSO) is a feature reduction technique that uses n-fold cross-validation to identify the features that explain the most variance in the data. For this implementation, we used n=10 fold cross validation to identify the fNIRS features that could serve as the best predictors. Overall, 24 fNIRS features were ultimately down-selected. These were fed into the model along with all 17 ECG, EDA, and respiration features.

IV. Problem Formulation

A. HMMs

Hidden Markov Models were initially chosen to tackle the problem of predicting the trust state from physiological observations such as ECG, EDA and also from fNIRS. HiddenMarkovModels.jl package is used for the implementation of the HMM models.

We have developed two models, a simpler HMM using just the physiological observations (ECG, EDA, RSP) and a full model involving a suite of 27 features from fNIRS and the mentioned physiological data.

Formally HMM can be represented as a tuple (S, O, T, E, π) as discussed in the previous sections.

- Hidden States (S): Binary classification of trust into High and Low trust.
- Observations (O): Simple model: ECG, EDA, RSP; Full Model: 27 fNIRS features + ECG, EDA, RSP.
- Transition Probability Matrix (T): Probability matrix for transitions between hidden states calculated as mentioned in the previous section.
- Emission Probability Matrix (E): a sequence of observation likelihoods or emission probabilities each representing a probability of an observation being generated from a state, calculated as previously discussed.
- Initial State Distribution (π): A probability distribution over the initial hidden states, this is taken as a uniform distribution. i.e, $[0.5, 0.5]$

B. Neural Networks

As will be discussed in later sections, difficulties were encountered in achieving good model performance with the HMM formalism and the Baum-Welch & Viterbi algorithms. An alternative formulation using neural networks was subsequently developed and explored. In this simplified formulation, the Markov property of trust dynamics was not exploited. Instead, neural networks designed to predict user trust directly from the current state's available physiological data were developed. We anticipated the performance of these models to exceed performance of the Baum-Welch and Viterbi algorithms since they can directly utilize ground-truth data in their training process. The network architectures are visualized in Fig. 6 and described below.

Simple Network:
Features: ECG, EDA, Respiration
Layers:

- Input Layer: 17 nodes, each with corresponding ECG, EDA, or Respiration feature
- Hidden Layer (x2): 34 nodes, densely-connected, leaky relu activation
- Output Layer: 1 node, absolute or differential trust prediction

Full Network:
Features: ECG, EDA, Respiration, and fNIRS
Layers:

- Input Layer: 41 nodes, each with corresponding ECG, EDA, Respiration, or fNIRS feature
- Hidden Layer (x2): 82 nodes, densely-connected, leaky relu activation
- Output Layer: 1 node, absolute or differential trust prediction

"Simple" and "Full" networks were developed and trained over 5000 iterations for each of the 4 participants. A "Simple" and “Full” network was also developed for the entire cohort and trained over 50,000 iterations. The best performing models were selected for evaluation against test data that was withheld from the training process. Additional details are provided in Section VI.B and C.

V. Solution Approach

The ”levels of success” we originally defined in the final project proposal evolved as we learned more about the HMM formalism and gained firsthand experience working with the HiddenMarkovModels.jl codebase. We believe the updated levels listed below are more ambitious, more aligned with the original intent of the project assignment, and more reflective of where our efforts as a group were directed.

A. Level 1: Minimum Working Example

Proposed Goal: Validate our implementation and understanding of the existing “HiddenMarkovModels.jl” package on data that is known to be describable with an HMM. Use the Baum-Welch algorithm to estimate the transition dynamics of an underlying HMM, then use the Viterbi algorithm to generate the maximum likelihood estimate for the underlying states of a given sequence of observations. Verify both algorithms’ implementations by comparing to previous work done in Python. (Unchanged from original proposal)

B. Level 2: Main Approach

Proposed Goal: Use the Baum-Welch algorithm from the HidddenMarkovModels.jl package to generate an HMM that models trust in an autonomous system based upon neurophysiological data.

Updated Goal: Generate **personalized, participant-specific** models of trust in autonomous systems using the Baum-Welch algorithm and a deep learning approach, and compare their performance.

Our overarching goal for the project was to develop a model that predicts human trust in autonomous systems based on a variety of neurophysiological data. We initially sought to generate this model solely through the use of the Baum-Welch algorithm. However, as we progressed with the model development and applied to Baum-Welch algorithm to our data, it became apparent that the Baum-Welch algorithm in its default form was not necessarily the best approach to generate a model of trust in autonomy on our specific data, since it does not incorporate ”ground truth” data in its calculations. Since all of the collected data in this project had associated ”truth” data in the form of participant surveys, an alternative approach using neural networks was conceived as a more viable alternative that is capable of improving its learning by incorporating truth data. Accordingly, we updated this goal to reflect our improved understanding of the best modelling approaches. Here, in addition to the original Baum-Welch proposal, we also present an alternative neural network-based approach to predicting user trust in autonomy and evaluate the performance of each approach.

C. Level 3: Reach Goal

Proposed Goal: Use the Viterbi algorithm from the HiddenMarkovModels.jl package to generate predictions of the most likely trust state sequences given observed neurophysiological data and a predicted HMM model generated via the Baum-Welch algorithm.

Updated Goal: Generate a **cohort, participant-agnostic** model of trust in autonomous systems using the best approach identified in the previous section.

Following the promising results achieved via neural network learning for personalized, individual trust-prediction models, we sought to develop a more general, participant-agnostic model of trust. We expected this goal to be much more difficult to achieve than the individual models previously discussed, given the appreciable inter-individual variability in neurophysiological signals and patterns. Our expectations were confirmed as we worked to develop these models.

VI. Results

A. Implementation Validation

It has been previously shown that the volatility in the price of gold can be modelled as a hidden Markov process, where the change in the daily price of gold is the observation, and the hidden state space is discretized into three states of “low,” “moderate,” and “high” volatility [13]. We used the HiddenMarkovModels.jl package to replicate the results of this previous work.

Gold data was accessed from gold.com and minor processing was conducted to ready the data for analysis. The initial HMM parameter estimates for use in the Baum-Welch algorithm (b, T, Z) are provided below

$$b = [0.9, 0.5, 0.5] \tag{11}$$

$$T = \begin{bmatrix} 0.8 & 0.15 & 0.05 \\ 0.1 & 0.8 & 0.1 \\ 0.05 & 0.1 & 0.8 \end{bmatrix} \tag{12}$$

$$Z = [\mathcal{N}(1,5), \mathcal{N}(1,10), \mathcal{N}(1,20)] \tag{13}$$

These initial guesses and data in the form of the daily change in gold price from Jan 1, 2008 through Sep 21, 2021 were passed to the HiddenMarkovModel.baum_welch algorithm, which returned the following optimized HMM parameters

$$b^* = [1, 0, 0] \tag{14}$$

$$T^* = \begin{bmatrix} 0.984 & 0.016 & 0 \\ 0.01 & 0.97 & 0.02 \\ 0.00 & 0.14 & 0.86 \end{bmatrix} \tag{15}$$

$$Z^* = [\mathcal{N}(0.25, 5.69), \mathcal{N}(0.30, 9.97), \mathcal{N}(0.04, 24.16)] \tag{16}$$

From the observation distributions Z^*, it can be seen that the algorithm successfully generated the dynamics for an HMM with low ($\sigma = 5.69$), moderate ($\sigma = 9.97$), and high ($\sigma = 24.16$) volatility states. These values match within numerical precision the values produced by the Python implementation.

Next, the optimized HMM parameters returned by the Baum-Welch algorithm were passed along with the original data into the Viterbi algorithm to predict the most likely underlying state sequence. The classification results are given in Fig. 2

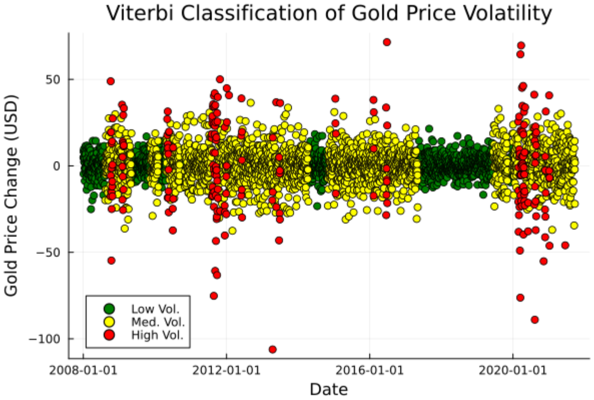

Fig. 2: Viterbi classification results using Baum-Welch-optimized HMM parameters.

For comparison, the classification results of the original work conducted in Python are given below in Fig. 3

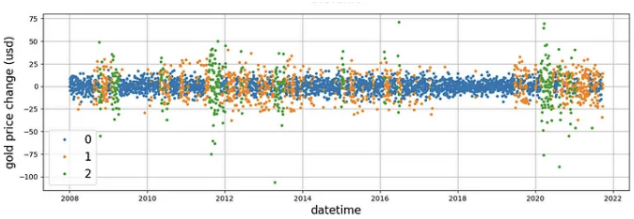

Fig. 3: Original Viterbi classification results using the Python implementation [13].

The quantitative agreement between the two Baum-Welch implementations and the visual qualitative agreements between the Viterbi classifications provide confidence that our usage of the HiddenMarkovModels.jl algorithms is correct and can be extended to the novel case of neurophysiological trust classification.

B. Personalized Models

The first model iteration aimed to predict trust in a personalized (i.e., n=1) model using all ECG, EDA, and respiration features as well as the down-selected fNIRS features from LASSO. We have opted to choose a binary trust model as a finer discretization of trust leads to algorithmic errors. This could be due to Baum-Welch's performance with the provided observation data or due to the in-house built method of initializing the HMM parameters θ. The results can be seen in Fig. 4

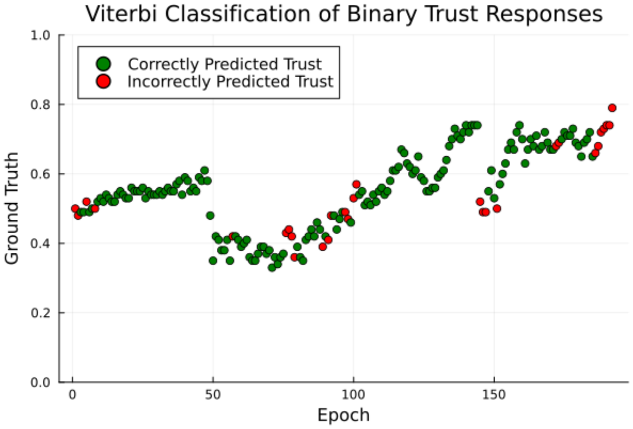

Fig. 4: Binary trust classification from a single subject using the Viterbi algorithm with fNIRS, respiration, ECG, and EDA features.

The second model iteration aimed to predict trust in a personalized model using ECG, EDA, and respiration features only. Although the sensors used in this experiment were research-grade, these are physiological sensors that are often employed in wearable devices to reduce their operational obtrusion (Can). Since the long-term goal of this research is employ trust prediction models in an operational environment, we want to compare the results from sensors that are less operationally obtrusive and less sensitive, to sensors that are more operationally obtrusive and more sensitive.

The initial observation distribution and initial transition probabilities are trained with the ground truth trust data. This information is fed into the Baum-Welch algorithm along with the selected feature set to obtain the learned transition and emission distributions. Then, these distributions and the feature set is fed into the Viterbi algorithm to predict the trust states of the participant. The results can be seen in Fig. 5.

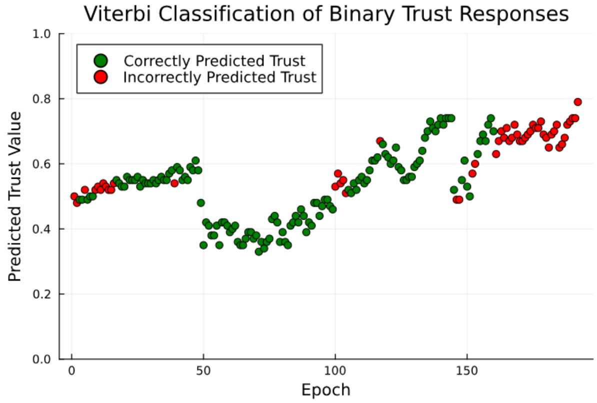

Fig. 5: Binary trust classification from a single subject using the Viterbi algorithm with respiration, ECG, and EDA features only.

The green dots in the figure denote the trust responses that were correctly predicted by the algorithm, whereas the red dots denote the trust responses that were incorrectly predicted by the algorithm. During the first 48 epochs (i.e., during the first session), the algorithm performed fairly well. The red dots seen here can most likely be attributed to the fact that they are all right around 50%, which is the cut off point for the binary low or high trust value. However, during the last 48 epochs (i.e., during the last session), most trust values were predicted incorrectly. This could be partly attributed to the poorer signal quality in the EDA data during session 4.

Given the difficulties we encountered in generating well-performing models via the Baum-Welch algorithm, we used a neural network as an alternative solution approach. The network architectures used are pictured below (Fig. 6). An input layer where each node corresponds to the value of a given neurophysiological feature, two hidden densely-connected layers with leaky relu activation whose sizes are twice that of the input layer, and a single output node containing the estimated trust value.

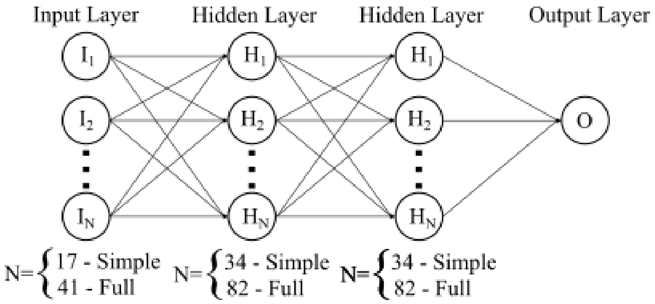

Fig. 6: Architectures of trust prediction networks used throughout this article.

Two neural networks of the architectures visualized in Fig. 6 were created and trained for each of the four participants. The "Simple" networks were provided only that participant's simplified set of neurophysiological features (ECG, EDA, and respiration parameters), while the "Full" networks were also provided a subset of 24 fNIRS features downselected from the original set of 180 features via the LASSO technique (described in Section V). Each participant's data was randomly split into a training and a test group, where the training groups received 80% (154 datapoints) and the test group received the remaining 20% (38 datapoints). The network parameters were trained on the training data using the ADAM stochastic gradient descent technique with step size 0.0005 implemented in the Flux.jl package [8]. Each network was trained over 5000 episodes, and the highest-performing "Simple" and "Full" networks (minimized MSE) for each participant were evaluated against the test data points. The performance of these selected networks is visualized in the form of cumulative error plots (Fig. 7). Not all models in Fig. 7 reach unity, as the output nodes of the networks were not constrained to the interval $[0, 1]$. It can be seen that 3 of the 4 participants' "Simple" models outperform the "Full" counterparts, and the 4^{th} participant's models have similar performance. While additional model tuning could ultimately reveal higher performing "Full" models, this result is surprising given recent literature suggesting that fNIRS data has higher associations with latent constructs like trust in autonomy [**?**]. Additional network architecture and training optimizations are necessary to improve performance of the "Full" model.

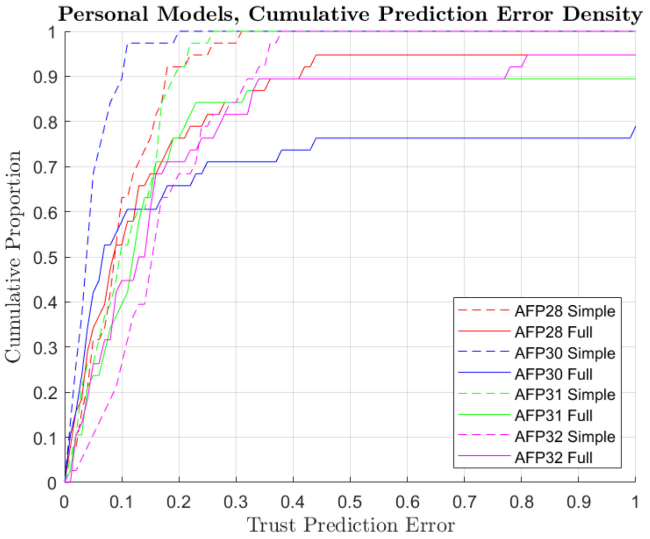

Fig. 7: Cumulative proportion of absolute trust prediction errors for participant-specific models using the simplified and full neurophysiological parameter suites. A point on a curve (x, y) represents the proportion y of test points whose predicted trust value was within x of the true trust value, akin to a cumulative distribution function.

In a similar vein, we used the same neural network framework discussed above to attempt to predict *changes* in trust from one timestep to the next. Eight individual models were generated (a "Simple" and "Full" model for each participant) under the same protocols discussed above, but instead were trained to predict the differential trust (i.e. instantaneous change in trust) between consecutive timesteps. Cumulative prediction error curves for each model are provided below in Fig. 8. These models generally performed marginally worse than the absolute trust prediction models given in Fig. 7, but overall performance is still quite high for 2 of the 4 participants. Interestingly, better agreement between the "Simple" and "Full" models is observed for these differential trust predictions than for the absolute trust predictions, which may suggest that the predictive potential of fNIRS data is stronger for future states than for the present state.

C. Cohort Models

Given the promising results uncovered for participant-specific trust prediction discussed in the previous subsection, we sought to extend this research direction to a generalized, participant-agnostic model of trust in autonomy. A well-performing, general model of trust prediction that can be applied to any individual would represent a major improvement

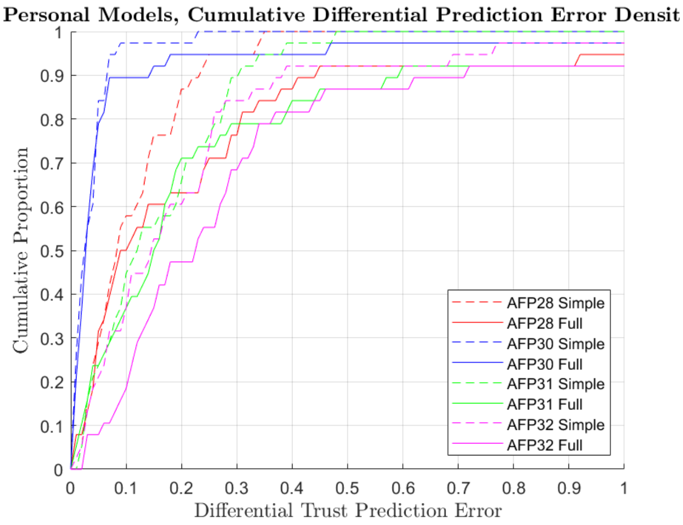

Fig. 8: Cumulative proportion of differential trust prediction errors for participant-specific models using the simplified and full neurophysiological parameter suites.

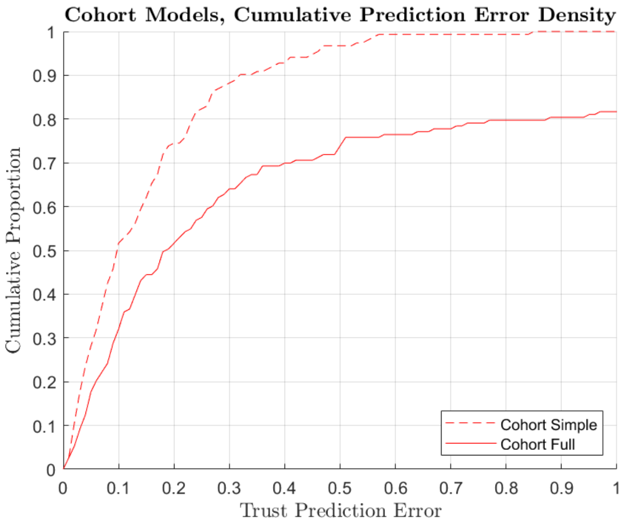

Fig. 9: Cumulative proportion of differential trust prediction errors for participant-agnostic models using the simplified and full neurophysiological parameter suites.

over current trust prediction schemes. Data from all four participants were combined and deidentified (i.e. no information regarding which particpant generated a given datapoint was provided to the network), split into a training and test group (again with 80%/20% splits), and used to train a “Simple” and “Full’ trust prediction network. These cohort models were trained by the same process described above but over 50,000 episodes (instead of 5000). The highest-performing models were evaluated against the test data. Cumulative prediction error plots akin to Fig. 7 are provided below in Fig. 9. As with the participant-specific models, the “Simple” model is seen to significantly outperform the “Full” model. Overall performance of the cohort model suffers relative to the participant-specific models, which is not unexpected given the significant inter-individual variability in neurophysiological signals.

VII. Conclusions

The strong performance of the simplified individual models for AFP28, AFP30, and AFP31 is quite promising and could inform future efforts to deploy similarly nonintrusive physiological sensors in operational contexts wherein operators are regularly interacting with autonomous systems. In predictions of both current and differential trust, we observe that the “Simple” models outperform the “Full” models. Future research efforts to operationalize these concepts may be better served by focusing their efforts on only the physiological data streams utilized in the “Simple” model as opposed to the full suite of neurophysiological data

While overall performance of the cohort models was generally unimpressive, even the moderate performance that was demonstrated is promising and suggests that such a generalize trust prediction model could be possible with more focused research. Improvements in cohort model performance could be realized by increasing the number of unique individuals whose data is included in the training process or by performing more robust feature down-selecting techniques.

VIII. Contributions and Release

The authors grant permission for this report to be posted publicly.

Contributions:

Joe Hesse-Withbroe:

- Created data processing codes to facilitate easy application of the algorithms contained in the HiddenMarkovModels.jl and Flux.jl packages to our data.
- Formulated and executed the neural network trust modeling approach.
- Generated all neural network data and cumulative error plots.
- Wrote Section IV.B, Section V, Section VI.A, the neural network results in Section VI.B and VI.C, and parts of Section VII.
- Managed the Git repo.

Sarah Leary:

- Responsible for human subject testing and pre-processing all data
- Executed minimum working example
- Tried out unique ways to ”train” the initial transition probabilities and observation distributions
- Trained HMM for the ”simple” personalized model
- Compiled and read the literature relevant to this project
- Wrote the abstract, Section I, Section III, the ”simple” personalized model results in Section VI.B

Pravesh Vadapalli

- Trained HMM for the ”full” personalized model
- Read about HMM algorithms and compiled Section II - basic formulation of HMM.

- Wrote the ”full” personalized model results in Section VI.B and Problem Formulation Section IV. A
- Executed minimum working example

Acknowledgment

The authors thank Ben Kraske for his helpful suggestions in formulating these models, as well as Prof. Zachary Sunberg and Benjamin Chupik for their support throughout the semester.

Appendix

Algorithm 1: Viterbi(states, init, trans, emit, obs)

```
Input: states: $S$ hidden states
Input: init: initial probabilities of each state
Input: trans: $S \times S$ transition matrix
Input: emit: $S \times O$ emission matrix
Input: obs: sequence of $T$ observations
$prob \leftarrow T \times S$ matrix of zeroes;
$prev \leftarrow$ empty $T \times S$ matrix;
for each state $s$ in states do
    $prob[0][s] = init[s] \times emit[s][obs[0]]$;
end
for $t = 1$ to $T - 1$ inclusive do
    for each state $s$ in states do
        for each state $r$ in states do
            $new\_prob \leftarrow prob[t-1][r] \times trans[r][s] \times emit[s][obs[t]]$;
            if $new\_prob > prob[t][s]$ then
                $prob[t][s] \leftarrow new\_prob$;
                $prev[t][s] \leftarrow r$;
            end
        end
    end
end
path $\leftarrow$ empty array of length $T$;
path$[T-1] \leftarrow$ the state $s$ with maximum $prob[T-1][s]$;
for $t = T - 2$ to $0$ inclusive do
    path$[t] \leftarrow prev[t+1]$[path$[t+1]$];
end
return path;
```

References

[1] K. Akash, W.-L. Hu, N. Jain, and T. Reid, “A Classification Model for Sensing Human Trust in Machines Using EEG and GSR,” ACM Trans. Interact. Intell. Syst., vol. 8, no. 4, pp. 1–20, Dec. 2018, doi: 10.1145/3132743.

[2] M. Benedek and C. Kaernbach, “A continuous measure of phasic electrodermal activity,” Journal of Neuroscience Methods, vol. 190, no. 1, pp. 80–91, Jun. 2010, doi: 10.1016/j.jneumeth.2010.04.028.

[3] S. Bhat, J. B. Lyons, C. Shi, and X. J. Yang, “Clustering Trust Dynamics in a Human-Robot Sequential Decision-Making Task,” IEEE Robot. Autom. Lett., vol. 7, no. 4, pp. 8815–8822, Oct. 2022, doi: 10.1109/LRA.2022.3188902.

[4] J. Chen and A. Schulte, “Special issue on ‘Human-Autonomy Teaming in Military Contexts,’” Hum.-Intell. Syst. Integr., vol. 3, no. 4, pp. 287–289, Dec. 2021, doi: 10.1007/s42454-021-00032-4.

[5] S. Hergeth, L. Lorenz, R. Vilimek, and J. F. Krems, “Keep Your Scanners Peeled: Gaze Behavior as a Measure of Automation Trust During Highly Automated Driving,” Hum Factors, vol. 58, no. 3, pp. 509–519, May 2016, doi: 10.1177/0018720815625744.

[6] S. K. Hopko and R. K. Mehta, “Neural Correlates of Trust in Automation: Considerations and Generalizability Between Technology Domains,” Front Neuroergon, vol. 2, p. 731327, Sep. 2021, doi: 10.3389/fnrgo.2021.731327.

[7] R. Hussain and S. Zeadally, “Autonomous Cars: Research Results, Issues, and Future Challenges,” IEEE Communications Surveys & Tutorials, vol. 21, no. 2, pp. 1275–1313, 2019, doi: 10.1109/COMST.2018.2869360.

[8] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization.” arXiv, Jan. 29, 2017. doi: 10.48550/arXiv.1412.6980.

[9] J. D. Lee and K. A. See, “Trust in Automation: Designing for Appropriate Reliance,” Hum Factors, vol. 46, no. 1, pp. 50–80, Mar. 2004, doi: 10.1518/hfes.46.1.50_30392.

[10] S. M. Merritt et al., “Automation-Induced Complacency Potential: Development and Validation of a New Scale,” Front Psychol, vol. 10, p. 225, Feb. 2019, doi: 10.3389/fpsyg.2019.00225.

[11] J. Moeyersons, M. Amoni, S. Van Huffel, R. Willems, and C. Varon, “R-DECO: an open-source Matlab based graphical user interface for the detection and correction of R-peaks,” PeerJ Comput Sci, vol. 5, p. e226, 2019, doi: 10.7717/peerj-cs.226.

[12] C. S. Nam and J. B. Lyons, Eds., “Chapter 1 - A multidimensional conception and measure of human-robot trust,” in Trust in Human-Robot Interaction, Academic Press, 2021, pp. 3–25. doi: 10.1016/B978-0-12-819472-0.09991-3.

[13] Y. Natsume, “Hidden Markov Models with Python. Modelling Sequential Data,” Medium.org. Accessed: May 07, 2024. [Online]. Available: https://medium.com/@natsunoyuki/hidden-markov-models-with-python-c026f778dfa7

[14] S. Palmer, D. Richards, G. Shelton-Rayner, D. Inch, and K. Izzetoglu, Human-Agent Teaming - an Evolving Interaction Paradigm: An Innovative Measure of Trust. 2019. Accessed: May 03, 2024. [Online]. Available: https://www.semanticscholar.org/paper/Human-Agent-Teaming-an-Evolving-Interaction-An-of-Palmer-Richards/f250455ceb2a703909ad9cd0ec25515a88f8dff0

[15] J. R. Perello-March, C. G. Burns, R. Woodman, M. T. Elliott, and S. A. Birrell, “Using fNIRS to Verify Trust in Highly Automated Driving,” IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 1, pp. 739–751, Jan. 2023, doi: 10.1109/TITS.2022.3211089.

[16] R. T. Scott et al., “Biomonitoring and precision health in deep space supported by artificial intelligence,” Nat Mach Intell, vol. 5, no. 3, Art. no. 3, Mar. 2023, doi: 10.1038/s42256-023-00617-5.

[17] J. Sung et al., “Operationally Realistic Human-Autonomy Teaming Task Simulation to Study Multi-Dimensional Trust,” in Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction, in HRI ’24. New York, NY, USA: Association for Computing Machinery, Mar. 2024, pp. 1028–1032. doi: 10.1145/3610978.3640693.

[18] S. S. Webber, “Development of Cognitive and Affective Trust in Teams: A Longitudinal Study,” Small Group Research, vol. 39, no. 6, pp. 746–769, Dec. 2008, doi: 10.1177/1046496408323569.

[19] H. M. Wojton, D. Porter, S. T Lane, C. Bieber, and P. Madhavan, “Initial validation of the trust of automated systems test (TOAST),” J Soc Psychol, vol. 160, no. 6, pp. 735–750, Nov. 2020, doi: 10.1080/00224545.2020.1749020.

[20] B. Yi et al., “How Can the Trust-Change Direction be Measured and Identified During Takeover Transitions in Conditionally Automated Driving? Using Physiological Responses and Takeover-Related Factors,” Hum Factors, p. 00187208221143855, Jan. 2023, doi: 10.1177/00187208221143855.

[21] B. Yi, H. Cao, X. Song, J. Wang, W. Guo, and Z. Huang, “Measurement and Real-Time Recognition of Driver Trust in Conditionally Automated Vehicles: Using Multimodal Feature Fusions Network,” Transportation Research Record, vol. 2677, no. 8, pp. 311–330, Aug. 2023, doi: 10.1177/03611981231156576.

[22] M. Zhao et al., “Teaching agents to understand teamwork: Evaluating and predicting collective intelligence as a latent variable via Hidden Markov Models,” Computers in Human Behavior, vol. 139, p. 107524, Feb. 2023, doi: 10.1016/j.chb.2022.107524.