Automatically Understanding Human Behavior for IoT Applications with Optimized Human-in-the-Loop Control

Himavanth Reddy Pallapu
Department of ECE
Stony Brook University
Stony Brook, NY, USA
himavanthreddy.pallapu@stonybrook.edu

Gnaneswar Villuri
Department of ECE
Stony Brook University
Stony Brook, NY, USA
ganeswar.villuri@stonybrook.edu

Alex Doboli
Department of ECE
Stony Brook University
Stony Brook, NY, USA
alex.doboli@stonybrook.edu

Simona Doboli
Department of Computer Science
Hofstra University
Hempstead, NY, USA
simona.doboli@hofstra.edu

***Abstract*—Modern IoT applications often address humans-in-the-loop (HiL) situations, which require that the automated control steps effectively and non-intrusively combine with the human decisions. This paper proposes a method to automatically characterize aspects of team behavior to improve HiL control in IoT applications. Experiments present the use of the method to optimize the distribution of the energy harvested through solar panels to teams based on the effectiveness of their behavior.**

***Index Terms*—IoT, humans-in-the-loop, teams, behavior modeling, power distribution**

I. INTRODUCTION

Modern IoTs often address situations in which human and machine decisions must be linked together, so that humans can benefit from automated, optimized decision making, while still having control over the outcomes of the application. This theoretical problem is sometimes referred to as Human - Machine alignment [1], and is a main Machine Learning (ML) challenge. A special category of IoTs involves small teams interacting through speech (audio signals) during a shared activity, like in collaborative design and problem solving (CDPS), healthcare, i.e. emergency rooms, power management, and operating complex industrial systems [2], [3].

The Humans-in-the-Loop (HiL) aspect [4] of such IoTs requires that the related control algorithms attempt to detect in a non-intrusive way not only the features of the pursued goals, like achieving the objectives as soon as possible or with the minimum amount of used resources (i.e. consumed energy), but also to identify human intentions, procedures, and steps, so that the automated and human decisions do not conflict with each other. As humans often interact through voice during a shared activity, the IoT control algorithms should attempt to track and understand the ongoing discussions, and incorporate the acquired information into their control steps.

While automatically understanding the complete semantics of human discussions is hard, this work argues that important insight can be extracted to achieve a better human - computer interaction. This paper describes a module to automatically understand facets of team behavior during CDPS for IoTs with HiL control. The behavior characterizations metrics and the HiL control equations are also presented. Experimental results discuss the use of the proposed module to describe team behavior to optimize the distribution of the limited energy harvested through solar panels. [5] explains that using human behavior models can help in significant energy savings.

This work relates to Department of Navy award N00014-22-1-2001 issued by the Office of Naval Research. The United States Government has a royalty-free license throughout the world in all copyrightable material contained herein. Simona Doboli's work was partially supported by Army Research Office Grant No. W911NF-20-1-0213.

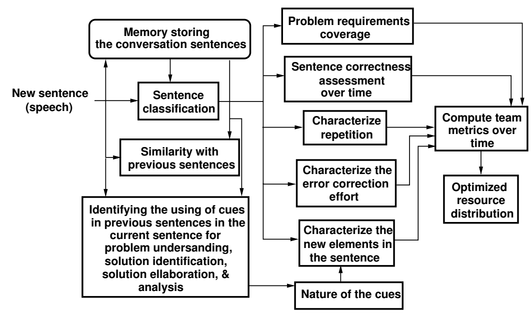

Fig. 1. Automatically understanding human behavior for IoTs with HiL

Section II presents the module to model team behavior. Section III focuses on a related application. Section IV details the algorithms followed by experiments and conclusions.

II. HUMANS-IN-THE-LOOP IN IOT APPLICATIONS

Figure 1 depicts the proposed process to model human decisions during team activities based on speech processing. Speaker diarization is first used to track human speech and associate the spoken ideas to the members of a team [6]. Then, the digitized audio signals are converted to text using an off-the-shelf speech-to-text algorithm. Each speech sentence is stored in memory, and input to three separate modules to identify the purpose of the sentence, to find its similarity (connection) to the previous sentences stored in the memory, and to identify the using of cues in previous sentences to formulate the current sentence. The three modules are described next:

- *Sentence classification*: It has been suggested that the sentences pertaining to a team activity [7], like CDPS,

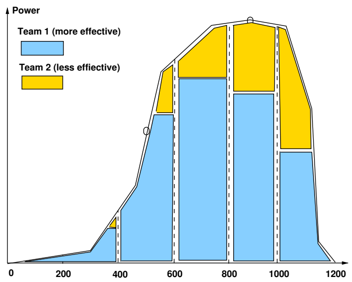

Fig. 2. The harvested solar energy and its distribution to two teams based on their behavior during CDPS

can be classified into five categories depending on their meaning: (i) problem understanding, i.e. requirements and assumptions, (ii) solution identification, like expressing the solution ideas, (iii) solution elaboration, e.g., implementing the idea, (iv) solution analysis about pros and cons, and (v) solution modification. Tracking the types of the spoken ideas describes the team's solving approach.

- *Sentence similarity*: The similarity of a sentence with the previous sentences indicates the degree to which a team pursues a given idea by proposing subsequent, related ideas (e.g., the convergence of the process) or it jumps to less related ideas (i.e. the divergence of the process).
- *Identifying the using of cues*: This module describes how cues in the previous sentences are used in the current sentence while considering the type of the sentence, as indicated by the classification module. For example, cues are used in question - answer situations, like a participant asks "`I don't know what void means`", and another participant responds "`A void means that basically the action has nothing`". The cue relating the two sentences is the word "`void`". Or, subsequent sentences might describe related solution elaborations, like one participant states that "`There is only one solution`", and another member indicates that "`The solution values are all positive`". The common cue is the word "`solution`". The nature of the cues is then characterized, like their frequency in sentences and the words they are related to, i.e. the associated verbs and nouns in the sentences.

The outputs of the three modules are used to calculate the metrics that characterize each team's behavior in terms of its capacity to produce correct solutions over time. The metrics relate to the degree to which the problem requirements are covered by the discussed solutions, the correctness of the sentences of a team conversation as an indirect measure of the solution correctness, the amount of repetitions, like repeated sentences as a measure of the amount of redundant work, the effort needed to correct solution errors and tackle unaddressed problem requirements, and the amount of novel concepts introduced at each step, as a measure of a team's capacity to think outside the box to address any needs. Section IV details the computing of the metrics. The metrics are then used to set-up the equations of the optimized control algorithms of an IoT application, like equations (1)-(6) in Sections III and IV.

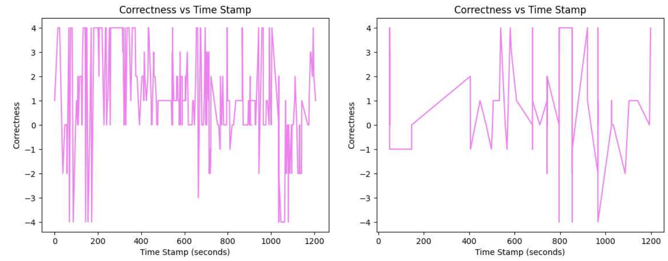

Fig. 3. Team behavior over time for a more effective team (left) and a less effective teams (right)

III. Application: Team behavior modeling for IoT in power grid management

This section illustrates an application of the module in Figure 1 to automatically understand human behavior for an IoT with HiL control. The application considers the IoT for solar power microgrid management, in which the available power is constrained, like autonomous teams that depend on portable solar energy harvesting, such as in various military conditions, or when the main power grid is unavailable, like medical teams that must function using local generators during natural disasters. Figure 2 illustrates the power profile of the energy harvested using solar panels. Other cases include the cold start of a power grid after a major power incident or a Cyber-attack. All these cases refer to CDPS examples for which the available power and energy is limited.

The metrics computed by the module for team behavior modeling offer insight into a team's effectiveness. Figure 3 plots the correctness of the sentences produced by a more effective team (left plot) and a less effective team (right plot). Note that the two plots show distinguishing features between the two teams, like the correctness of the spoken sentences, hence the related solutions, is high and consistently, early in the CDPS process, for the effective team, but becomes high only later and for short bursts for the less effective team. The metrics (and their distinguishing features, as in Figure 3) set-up a model for the expected team success in the CDPS activity. Section IV presents the metrics and algorithms. Section V discusses the related experiments. The harvested but limited energy is distributed to teams based on their expected success. The next three equations control the energy distribution.

$$\sum_{i}^{\#teams} \sum_{j}^{T} P_i^{(Req)}(Act\ \ (S_j^{(i)}(t)))\ \ dt \ \le \ \int^{T} P^{(Gen)}\ \ dt \qquad (1)$$

Equation (1) states that the total energy required by the participating teams should be less or equal to the generated energy, e.g., the solar energy harvested in time T. For each team i, the required power P depends on the conducted activity Act as described by a sentence $S_j^{(i)}$. For example, the discussion about elaborating a solution requires adding details to an implementation, which then requires a specific amount of energy. However, the activities during solution analysis require running an extensive tests to characterize the errors and performance of a solution, which usually utilizes more energy than the energy needed for solution elaboration.

$$\max \sum_{i}^{\#teams} \sum_{j}^{T} Correctness\ (S_j^{(i)}(t)) \tag{2}$$

Equation (2) describes the first optimization criterion, which is to maximize the total number of correct steps performed by the participating teams during the time interval T. The function $Correctness$ ranks the correctness of the sentences $S_j^{(i)}$.

$$\min \sum_{i}^{\#teams} \sum_{j}^{T} Correctness\ (S_j^{(i)}(t))\ Delay\ (S_j^{(i)}(t))\ dt \tag{3}$$

Equation (3) states the second optimization criterion, which is that the delay in a team's activity due to an unavailable activity is less for activities with a higher degree of correctness.

Figure 2 illustrates the distribution of the harvested solar energy to the two teams in Figure 3: the energy distribution to the more effective team is shown in blue color and to the less effective team is depicted in yellow color.

IV. Related Algorithms

This section details the algorithms of the module to automatically understand the human behavior in IoT applications with HiL, and shown in Figure 1. The modules compute a set of metrics, which are then used to set up the equations used to express the optimization goals of the IoT control algorithms.

For sentence classification in Figure 1, our work experimentally evaluated fifteen state-of-the-art, off-the-shelf classification algorithms. The algorithms are summarized in Table I. Each classifier was trained to classify a sentence depending on its semantics into the five categories enumerated in Section II. The classifier training used the verbal discussions of 27 teams that participated to a CDPS activity of twenty minutes. As discussed in Section V, results suggested that BERT transformer model offers the best accuracy in distinguishing the spoken sentences depending on their meaning for the CDPS process.

The sentence similarity module computes the similarity of two sentences by first tokenizing the sentences and removing the stop words. Then, the term frequency vectors are calculated for each word, and a combined vector of all unique terms is formed. Next, the cosine similarity between the two vectors is found using the formula $cosine = \frac{A\ B}{||A|| \times ||B||}$, where A and B are the two term frequency vectors. The cosine similarity is the measure of the similarity between the two sentences.

The next metrics are calculated for each classified sentence:

1) *The likelihood of correct and incorrect understanding of the problem requirements*: The sentences classified as problem understanding are assessed about their coverage of the problem requirements. This is done by matching the nouns connected through verbs in the spoken sentences to the nouns and their relations through verbs in the problem description. Unaddressed requirements are flagged. Also, the correctness and errors of the sentences in this category are assessed over time. This assessment is currently manual, but we plan to train a DNN model.
2) *The new elements introduced by a sentence*: The metrics describe the number of new solution concepts (nouns and verbs) added by a sentence on solution identification and elaboration when building a correct implementation.
3) *The likelihood of introducing errors during solution identification and solution elaboration*: Similar but separate metrics are computed for the ideas classified as part of solution identification and the ideas pertaining to solution elaboration. The metrics count over time the number of errors introduced by each sentence, the degree to which the sentences correctly relate to the problem description and problem understanding mentioned by Item 1, and the repetition of the same errors.
4) *The error correction effort*: Separate metrics are computed for the spoken ideas on problem understanding, solution identification, and solution elaboration to describe the steps it took to fix an error. The distribution of the sentence types (the five types in Section II) is also found between the sentence that introduces an error and the sentence that fixes the error.
5) *The likelihood of idea repetition*: Separate metrics are computed for each sentence type to characterize the amount of content repetition. Repetitions during team discussions arguably indicate redundant steps, and can possibly indicate inefficiencies during the team activity.

In addition to equations (1)-(3), the following equations defined using the above metrics describe the HiL control algorithm of an IoT application:

$$\#_{steps}^{(i)} = \#_{bridge\ gap}^{(i)} + \#_{correct\ errors}^{(i)} \tag{4}$$

The equation states that the expected number of steps required by team i to solve a problem, $\#_{steps}^{(i)}$, includes the expected number of steps required to bridge the gap between the problem requirements and the solution implementation, $\#_{bridge\ gap}^{(i)}$, plus the number of steps needed to correct any errors made in the solving process, $\#_{correct\ errors}^{(i)}$.

The expected number of steps $\#_{bridge\ gap}^{(i)}$ to bridge the gap between the problem requirements and the solution implementation can be expressed as follows:

$$\#_{bridge\ gap}^{(i)} = \frac{Gap_{problem-impl}}{\Delta_{New,impl}^{(i)}} (1 + prob_{repetition}^{(i)}) \tag{5}$$

Parameter $Gap_{problem-impl}$ is the expected knowledge gap between the problem requirements and implementation (Item 1 in the above list). This gap must be covered during CDPS. Parameter $\Delta_{New,impl}^{(i)}$ is the expected amount of new elements introduced by each sentence (Item 2), and parameter $prob_{repetition}^{(i)}$ is the likelihood of idea repetition (Item 5).

The expected number of steps $\#_{correct\ errors}^{(i)}$ to correct all errors of an implementation is as follows:

$$\#_{correct\ errors}^{(i)} = \#_{bridge\ gap}^{(i)}\ prob_{errors}^{(i)}\ Effort_{error}^{(i)} \tag{6}$$

Parameter $prob_{errors}^{(i)}$ is team i's likelihood to introduce an error during each CDPS step (Item 3), and parameter $Effort_{error}^{(i)}$ is the expected effort to correct an error (Item 4).

The HiL control algorithm of an IoT application solves equations (1)-(6) set-up for each team sharing a joint resource, like the available energy amount, to produce energy distributions, like that shown in Figure 2.

TABLE I
PERFORMANCE OF TRADITIONAL CLASSIFICATION ALGORITHMS

	Performance Metrics							
	Accuracy (%)				Precision (%)			
	Min	Avg.	Max	Std. dev.	Min	Avg.	Max	Std dev.
DTC	26	47	63	8	25	50	61	8
GOSDT	24	37	54	7	18	41	58	7
CONST.DTC	45	54	68	5	41	52	71	7
LR	37	50	70	8	36	52	75	9
RFC	46	58	77	7	35	60	76	10
KNN	46	58	77	7	35	60	76	10
SVM	36	53	72	9	20	28	80	16
NB	35	46	66	8	12	46	64	18
LDA	30	40	52	8	23	55	75	13
AdaBoost	36	50	60	8	35	51	77	10
GBC	42	54	62	6	45	60	80	8
FNN	54	60	72	5	49	58	74	6
BERT	51	66	78	6	48	65	74	9
DistilBERT	48	64	74	8	42	60	71	9
RoBERTa	49	53	66	5	38	57	63	13

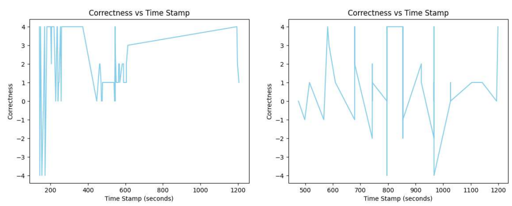

Fig. 4. Likelihood of correct and incorrect solution implementation ideas for a more effective team (left figure) and for a less effective team (right figure)

V. EXPERIMENTAL RESULTS

Experiments first considered a broad set of existing, off-the-shelf classification methods to separate the ideas expressed during team CDPS into the five categories enumerated in Section II. Table I summarizes the results by indicating the accuracy and precision of the algorithms. Each row corresponds to a different classifier. The analyzed algorithms were as follows: Decision Tree Classifier (DTC) [9], GOSDT DTC (GOSDT) [8], a mathematically optimal DTC for a given number of leaf nodes, the DTC in row 1 with its number of leaves constrained to be the same as the number of leaves in GOSDT (CONST.DTC), Logistic Regression Classifier (LR) [9], Random Forest Classifier (RFC) [9], K-Nearest Neighbor Classifier (KNN) [9], Support Vector Machine Classifier (SVM) [9], Naive Bayes Classifier (NB) [9], Linear Discriminant Analysis (LDA) [9], AdaBoost Classifier (AdaBoost) [9], Gradient Boosting Classifier (GBC) [9], Feedforward Neural Network (FNN) [10], BERT [11], DistilBERT [12], and RoBERTa [13]. The experimental results show that BERT classifier offers the best accuracy with a minimum, average, and maximum accuracy of 51%, 66%, and 78% respectively, and a standard deviation of only 6%. The classifiers' recall and F1-score were also found, and similar trends were observed.

Figure 3 shows the overall correctness over time of the CDPS steps of a more and a less effective team, while Figure 4 illustrates the correctness only of the implementation steps for the two teams. Note that the effectiveness of behavior of the two teams can be distinguished based on these metrics.

Figure 5 depicts the error correction effort for a more and

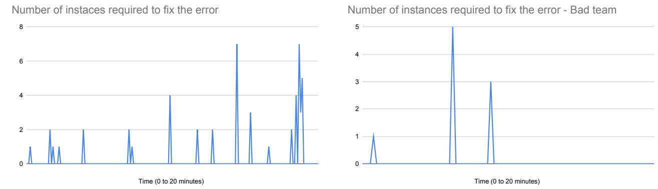

Fig. 5. Error correction effort for a more effective team (left figure) and for a less effective team (right figure)

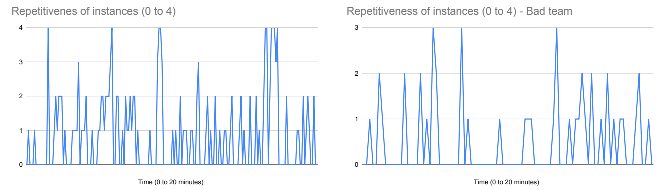

Fig. 6. Sentence repetitiveness for a more effective team (left figure) and for a less effective team (right figure)

a less effective team, e.g., the number of steps required to fix a previous error. Note that the more effective team addresses more errors in less steps, while the less effective team could fix only three errors. Figure 6 indicates the repetitiveness of sentences as an indirect measure of the redundant work. Surprisingly, the more effective team had a higher idea repetition than the less effective team.

VI. CONCLUSIONS

This paper presents a method to automatically characterize the team behavior to improve HiL control in IoT applications. Modern IoT applications often tackle HiL requirements. Experiments discuss optimizing the distribution of the harvested solar energy to teams based on their behavior effectiveness. Future work will tackle other IoT applications with HiL.

REFERENCES

[1] B. Christian, "The alignment problem : machine learning and human values", *W.W. Norton & Company*, 2020.
[2] Q. Liu *et al.*, "Surrogate-Assisted Many-Objective Optimization of Building Energy Management", *IEEE Computational Intelligence Magazine*, 18(4), pp. 14–28, 2023.
[3] A. Stefanini *et al.*, "Patient satisfaction in emergency department: Unveiling complex interactions by wearable sensors", *Journal Business Research*, 129, pp. 600-611, 2021.
[4] W. Li *et al.*, "Synthesis for Human-in-the-Loop Control Systems", In: E. Abraham, K. Havelund (eds), *Tools and Algorithms for the Construction and Analysis of Systems*, TACAS, Lecture Notes in Computer Science, vol 8413, 2014.
[5] S. Munir *et al.*, "Reducing Energy Waste for Computers by Human-in-the-Loop Control", *IEEE Trans. Emer. Topics Computing*, 2(4), 2013.
[6] R. Duke and A. Doboli, "diaLogic: Non-Invasive Speaker-Focused Data Acquisition for Team Behavior Modeling", *arXiv*, 2209.00619, 2022.
[7] R. Duke and A. Doboli, "Applications of diaLogic System in Individual and Team-based Problem-Solving Applications", *IEEE International Symposium on Smart Electronic Systems (iSES)*, 2022.
[8] J. Lin *et al.*, "Generalized and scalable optimal sparse decision trees", *Proc. International Conference on Machine Learning*, 571, 2020.
[9] https://scikit-learn.org/stable/modules/generated/
[10] https://github.com/fchollet/keras.
[11] https://huggingface.co/docs/transformers/model_doc/bert.
[12] https://huggingface.co/docs/transformers/model_doc/distilbert.
[13] https://huggingface.co/docs/transformers/model_doc/roberta.