DEEPSTEALTH: Game-Based Learning Stealth Assessment with Deep Neural Networks

Wookhee Min, Megan H. Frankosky, Bradford W. Mott, Jonathan P. Rowe,
Andy Smith, Eric Wiebe, Kristy Elizabeth Boyer, and James C. Lester

***Abstract*—A distinctive feature of game-based learning environments is their capacity for enabling stealth assessment. Stealth assessment analyzes a stream of fine-grained student interaction data from a game-based learning environment to dynamically draw inferences about students' competencies through evidence-centered design. In evidence-centered design, evidence models have been traditionally designed using statistical rules authored by domain experts that are encoded using Bayesian networks. This article presents DEEPSTEALTH, a deep learning-based stealth assessment framework, that yields significant reductions in the feature engineering labor that has previously been required to create stealth assessments. DEEPSTEALTH utilizes end-to-end trainable deep neural network-based evidence models. Using this framework, evidence models are devised using a set of predictive features captured from raw, low-level interaction data to infer evidence for competencies. We investigate two deep learning-based evidence models, long short-term memory networks (LSTMs) and *n*-gram encoded feedforward neural networks (FFNNs). We compare these models' predictive performance for inferring students' knowledge to linear-chain conditional random fields (CRFs) and naïve Bayes models. We perform feature set-level analyses of game trace logs and external pre-learning measures, and we examine the models' early prediction capacity. The framework is evaluated using data collected from 182 middle school students interacting with a game-based learning environment for middle grade computational thinking. Results indicate that LSTM-based stealth assessors outperform competitive baseline approaches with respect to predictive accuracy and early prediction capacity. We find that LSTMs, FFNNs, and CRFs all benefit from combined feature sets derived from both game trace logs and external pre-learning measures.**

***Index Terms*—Computational Thinking, Deep Learning, Educational Games, Game-Based Learning, Stealth Assessment**

I. INTRODUCTION

Recent years have seen growing interest in intelligent game-based learning environments because of their potential to create personalized and engaging learning experiences [1]. These environments simultaneously merge adaptive pedagogical functionalities delivered through intelligent tutoring system functionalities with the engaging learning experiences provided by digital games [2], [3], [4], [5]. Recent work on game-based learning has explored a broad spectrum of subject matter ranging from K-12 mathematics [4], [6], elementary school social behaviors [7], middle school computer science [8], anti-bullying [9], social language and culture learning [3], science inquiry [10], and biosafety training [11].

A key benefit of game-based learning environments is their ability to embed problem-solving challenges within interactive virtual environments, which can enhance students' motivation [1], [12]. These environments facilitate learning through customized narratives, feedback, and problem-solving support [13], [14], [15]. Game-based learning environments are a promising laboratory for a wide range of artificial intelligence-driven student modeling efforts to infer development of competencies [14], [16], study affective states centering around learning [17], [18], and monitor progression towards learning goals [19] by utilizing fine-grained streams of students' interaction data that represent problem-solving behaviors.

A significant design challenge posed by intelligent game-based learning environments is understanding how to robustly measure learning without disrupting engagement. *Stealth assessments* address this challenge by embedding unobtrusive assessments within game mechanics, offering a real-time non-disruptive assessment method [14]. Stealth assessment examines student interaction data to provide real-time behind-the-scenes measurement of students' learning processes and outcomes [16], [20]. Specifically, students' learning is inferred by analyzing detailed sequences of observed behavioral cues that indirectly reveal competencies for knowledge and skills without conducting explicit formative assessments. This information can be utilized to provide tailored problem-solving support for individual learners in a way that is both timely and

Manuscript received September 10, 2018; revised March 13, 2019; accepted May 25, 2019. This work was supported by the National Science Foundation under Grants CNS-1138497 and DRL-1640141. *(Corresponding author: Wookhee Min.)*

W. Min is with the Department of Computer Science, North Carolina State University, Raleigh, NC 27695 USA (e-mail: wmin@ncsu.edu).

M. H. Frankosky is with the Intelligent Devices Group at Lenovo, Morrisville, NC 27560 USA (e-mail: meganfrankosky@gmail.com).

B. W. Mott is with the Department of Computer Science, North Carolina State University, Raleigh, NC 27695 USA (e -mail: bwmott@ncsu.edu).

J. P. Rowe is with the Department of Computer Science, North Carolina State University, Raleigh, NC 27695 USA (e-mail: jprowe@ncsu.edu).

A. Smith is with the Department of Computer Science, North Carolina State University, Raleigh, NC 27695 USA (e-mail: pmsmith4@ncsu.edu).

E. Wiebe is with the Department of STEM Education, North Carolina State University, Raleigh, NC 27695 USA (e-mail: wiebe@ncsu.edu).

K. E. Boyer is with the Department of Computer & Information Science & Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: keboyer@ufl.edu).

J. C. Lester is with the Department of Computer Science, North Carolina State University, Raleigh, NC 27695 USA (e-mail: lester@ncsu.edu).

contextually appropriate [20], [21]. It can also inform teachers of potential pedagogical adaptations or integration with additional curricular activities, which are key components of distributed and integrated scaffolding [22], [23], [24].

Stealth assessment is methodologically grounded in evidence-centered design (ECD), a process for designing valid knowledge assessments [25]. ECD features task, evidence, and competency models for diagnostic measurement of multiple aspects of students' proficiency and performance. Built on the three models presented in ECD, stealth assessments utilize a rich stream of student interactions (i.e., an evidence model) collected from various problem-solving tasks (i.e., a task model) in game-based learning environments to draw inferences about student knowledge and skills (i.e., a competency model). The evidence model provides the connections between the competency model and the stream of low-level observations from student interactions with the task, enabling the competency model to update competency variables in the respective topic or skill [20]. Real-time processing of these three models points the way toward intelligent, adaptive game-based learning environments with development of robust evidence models being a key goal.

Developing stealth assessments is a labor-intensive process requiring highly skilled subject matter experts to manually design reliable evidence and competency models to accurately infer student knowledge and skills. This typically demands stealth assessment designers to undertake manual feature engineering efforts and design probabilistic graphical models (e.g., [14], [20], [26]). As an approach to reducing this cost, we present DEEPSTEALTH, a stealth assessment framework that leverages *deep learning* (DL) for automatically devising evidence models [16]. DL is a family of machine learning algorithms that utilize deep neural networks to automatically extract hierarchical features from low-level data (e.g., a sequence of student behaviors in a game-based learning environment) [27]. DEEPSTEALTH has shown significant promise for alleviating the expensive and labor-intensive process of designing evidence models [8], [16]. Findings indicate that an evidence model implemented as a long short-term memory network, which is a particular type of DL architecture, outperforms an *n*-gram encoded feedforward neural network, an alternative type of DL architecture, as well as non-DL models that were induced using expert-engineered features [8]. This current work further investigates the capabilities of DEEPSTEALTH focusing on three key research questions (RQs):

RQ1. Can DEEPSTEALTH-based evidence models outperform other competitive approaches with respect to predictive accuracy when models are trained using only raw, low-level action sequences along with external pre-learning measures?

RQ2. Which features of game interaction logs and external learning measures serve as strong predictors for evidence modeling with respect to predictive accuracy?

RQ3. Can DEEPSTEALTH evidence models outperform other competitive approaches with respect to *early prediction*?

To answer RQ1, we examine four computational methods including two DEEPSTEALTH models (long short-term memory networks and *n*-gram encoded feedforward neural networks), conditional random fields (probabilistic models dealing with sequential inputs), and *n*-gram encoded naïve Bayes models (probabilistic models dealing with fixed size inputs), where the input for these models are pre-learning measures and low-level sequences of student actions instead of engineered features, and the output of the models is evidence for one of the core competencies in a computational thinking curriculum.

To address RQ2, we investigate the independent influence of the *game interaction log* feature set (i.e., action-level student behaviors in our game-based learning environment) and the *external pre-learning measure* feature set (i.e., content knowledge pre-test scores, self-efficacy questionnaire scores [28], and self-reported computer science attitudes [29] measured prior to gameplay). We evaluate the predictive capacity of the two independent feature sets by devising two distinct evidence models per computational approach. We compare these two models to a combined model that utilizes both feature sets together. This feature set-level analysis investigates how different machine learning techniques handle data from two different modalities.

To address RQ3, we evaluate the early prediction capability of the four computational methods. Early prediction is particularly important in game-based learning environments because run-time adaptive scaffolding is a central objective of stealth assessment. We measure models' early prediction capacity using *standardized convergence point*, which is a metric that estimates how early predictions converge to the correct competency level in each sequence [30]. For this metric, a lower score is more desirable since it indicates that model predictions converge to the correct label sooner.

This article is organized as follows. Section II presents related work on intelligent tutoring systems and stealth assessment. Section III describes ENGAGE (Figure 1), a game-based learning environment for computational thinking targeted in middle school, which is used as a testbed environment for DEEPSTEALTH. Section IV describes the student data corpus and instruments administered in multiple classroom studies across four public middle schools in the southeastern United States. Section V introduces the DEEPSTEALTH framework, and Sections VI and VII present empirical results centering on the three research questions along with a discussion of the findings. Finally, the article concludes with directions for future work.

II. RELATED WORK

A. Intelligent Tutoring Systems

Intelligent game-based learning environments are situated at the intersection of (1) digital games that increase students' motivation through rich settings, engaging characters, and compelling plots in virtual environments, and (2) intelligent tutoring systems (ITSs) that foster students' learning through tailored scaffolding and context-sensitive feedback [1]. A rich body of work on ITSs has explored a broad range of computational approaches for student knowledge modeling, particularly inferring competencies in knowledge and skills using observed sequences of performance on tasks. Examples

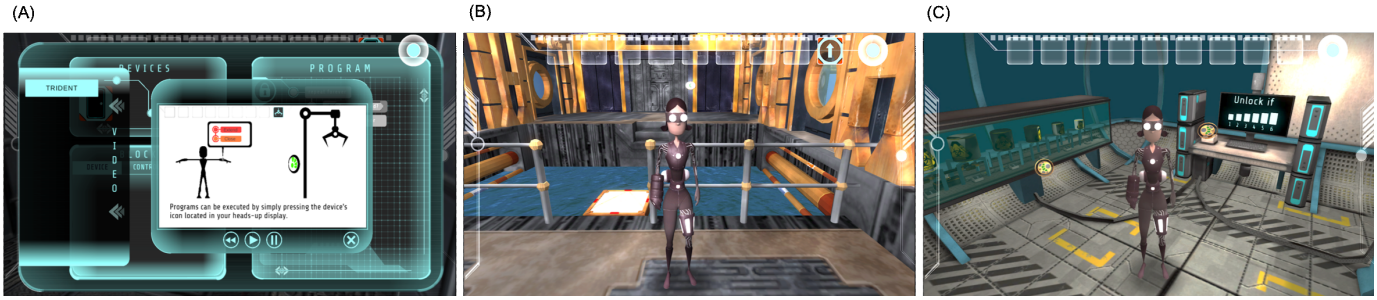

Fig. 1. Screenshots from the ENGAGE game-based learning environment: (A) an instructional video that explains how to run a device program, (B) a moving platform task in the Introduction level, and (C) a bubble-sort task in the Big Data level.

include factor analysis techniques, such as learning factors analysis [31], performance factors analysis [32], [33] and matrix/tensor factorization [34], [35], [36], which have been investigated for modeling latent knowledge states based on student performance on exercises. Item response theory (IRT) adopts a logistic function to model the probability of correctly answering an exercise [37]. For example, one variant of IRT models features three parameters: the difficulty of an exercise, the random guess, and the discrimination, in which the probability is inferred depending on the student's skill level associated with the exercise.

Bayesian knowledge tracing (BKT) assesses students' latent knowledge and skills in the context of cognitive modeling to predict their performance on future exercises [38]. Based on hidden Markov models, standard BKT models aim to predict students' latent knowledge utilizing four parameters: the initial probability of knowing a skill *a priori*, the probability of transitioning knowledge of a skill from unknown to known, the probability of a slip (i.e., making a mistake when applying a known skill), and the probability of a guess (i.e., successfully applying a skill without having mastered it). Parameter values can be fit using optimization techniques such as expectation maximization and conjugate gradient search [39]. Individualized BKT models that consider learner-specific aspects such as initial probability of mastery [40], speed of learning [39], and student-based parameter fit [41] have demonstrated improved predictive performance compared to classical BKT approaches.

More recently, deep knowledge tracing (DKT) has demonstrated an approach to knowledge tracing that uses recurrent neural networks [42]. Instead of requiring hand-crafted model parameters as well as labeling a skill for each exercise, DKT takes as input a sequence of a student's exercise results (i.e., correctness of exercises) in order to predict the probability of answering the next exercise correct at the following time step, thereby exhibiting improved scalability for student knowledge modeling compared to BKT.

B. Stealth Assessment

Evidence-centered design (ECD) is an assessment framework that harnesses evidentiary arguments to connect task-level evidence (e.g., what students do, say, or create) with higher-level skills and knowledge concepts in order to infer ones' competencies [25], [43], [44]. Specifically, the conceptual assessment framework in ECD defines three operational models, centering around what is being learned (competency model), where the knowledge is being demonstrated (task model), and how to connect the two models (evidence model), which together can be used to deliver student-adaptive learning content and feedback [43].

Stealth assessment extends ECD to game-based assessment [26]. Student interactions with game-based learning environments produce fine-grained evidence in the form of raw game trace logs, such as a history of places that the students have visited, interactions with non-player characters, and a sequence of steps taken to solve a task situated within the learning environment.

Various families of machine learning techniques have been investigated for evidence modeling in these environments. Kim and colleagues investigated Bayesian network-based evidence modeling, which requires two primary steps: (1) defining targeted competency and observable variables and building a directed graphical model, and (2) specifying the conditional probabilities between parent nodes and corresponding child nodes [26]. Falakmasir et al. presented the SPRING data analysis pipeline that does not require costly domain knowledge engineering [45]. Specifically, SPRING trains two hidden Markov models (HMMs), one for high-performing and the other for low-performing students per game level. Two log-likelihoods of an observed sequence of student events are computed based on the two HMMs, and the difference between the two log-likelihoods for each game level is used as an independent variable for a linear regression model that predicts post-test scores.

III. ENGAGE GAME-BASED LEARNING ENVIRONMENT

To investigate deep learning-based evidence models for stealth assessment, we utilize a game-based learning environment designed to introduce computational thinking to middle school students, ENGAGE (Figure 1). ENGAGE features a rich immersive 3D storyworld built with the Unity multi-platform game engine and Flare user interface toolkit [46]. The curriculum underlying ENGAGE is based on the AP Computer Science Principles course [47] with adapted learning objectives that are developmentally appropriate for U.S. middle school students (ages 11-13).

Computational thinking is an approach to problem solving that involves several key practices, including abstracting, algorithmic thinking, systematic information processing, and leveraging computational tools for data analysis, modeling, or

TABLE I
DESCRIPTIONS OF ENGAGE'S THREE GAME LEVELS

Levels	Key Concepts to Learn	CS Principles Objective Statements
Introduction	• Game mechanics (e.g., controlling player character, using the visual programming language) • Introductory programming skills and interactions with gameworld devices (e.g., moving platforms, cranes)	• Programming languages are a tool through which people implement algorithms to solve problems using their creativity and skills.
Digital World	• The concept that binary numbers can represent various types of data such as decimal numbers, alphabetical characters, and colors • Intermediate programming skills (e.g., iteration, conditionals, data conversion) using various gameworld devices (e.g., binary locks/lifts for numbers, floor tiles for colors and alphabetical characters)	• Binary is an abstraction that computers use to communicate, and the meaning of any binary sequence will depend on its interpretation and use.
Big Data	• Data analysis including filtering, sorting, visualizing, and discovering empirical findings from the analysis • Advanced programming skills that require programming based on computational thinking (e.g., developing algorithms) using various gameworld devices (e.g., bubble sorting device, screen devices for filtering, sorting, and visualization)	• People use computers to analyze data and discover new information with practical applications to real-world problems.

simulations [48], [49]. The problem-solving challenges within ENGAGE were designed to develop computational thinking skills and strategies through the creation and analysis of computational artifacts. In addition to focusing on development of computational thinking strategies, these challenges also aim to increase interest in computer science and provide a foundation for more advanced computer science work in high school.

In ENGAGE, students play the role of the protagonist who has been sent to an underwater research facility to restore its communication systems, which have been sabotaged by a non-player villain character. As students explore the research facility, they progress through each level of the game, which consists of a series of interconnected rooms. Each room presents students with a set of computational challenges students must solve by either programming devices located in the room or interacting with devices to appropriately execute written programs. To program devices, students use a visual programming interface to drag and drop "blocks" that represent functional units to create programs that contain an ordered series of commands and controls to be executed by the device. Programming in ENGAGE is inspired by the Scratch visual programming environment [50]. As students develop block-based programs, scaffolding is provided through brief instructional videos (Figure 1A) and non-player character dialogue, which unfolds using animated vignettes sequenced in three thematic levels: Introduction (Figure 1B), Digital World (Figure 2), and Big Data (Figure 1C). Table I introduces key learning concepts and the Computer Science Principles Objective Statement associated with each level. Each level was iteratively refined and developed through a series of curriculum design activities with middle school teachers and students based on the AP Computer Science Principles course [47]. Through a series of highly interactive learning activities, students repair the communication systems and stop the villain from causing further harm on the research station.

Several studies have investigated student learning in the ENGAGE game-based learning environment. One thread of research investigates how to achieve gender equity in ENGAGE with respect to learning gains through collaboration [51] and levels of frustration through a learning companion [52]. Students who interacted with ENGAGE and had no prior programming experience increased their confidence subscale of the computer science attitudes survey [29] to nearly the level of those who also interacted with ENGAGE but came with prior programming experience [53]. Frankosky and colleagues conducted a latent class growth analysis on students' interactions on six programming challenges within the ENGAGE game [54]. They identified three distinct groups of students: (1) the "steady performance" group (consistently spending less programming time than average), (2) the "quickly improving" group (after spending higher than average time for the first two challenges, trending rapidly downward in programming time), and (3) the "gradual lag" group (exhibiting higher than average programming time). In addition, students' interaction trace data have been analyzed to infer their learning outcomes in the context of stealth assessment [8], [16], [55].

In this work, we focus on students' problem-solving activities within ENGAGE's Digital World level, in which students investigate how binary sequences are used to represent digital data. In problem-solving activities, students find the binary representation of a base-ten number to activate a lift device (Figure 2, Left), which requires them to review an existing program for the lift device (Figure 2, Right) to determine what base-ten number activates the lift. Students thereby gain an understanding of the concept of bits in binary numbers and the weight assigned to each bit. Students first pair the lift device, which is the process of registering a device with their virtual in-game computer in order to manipulate or view a device's program. Then, students read the program using the visual programming interface, and flip binary tiles on the lift device (e.g., the white squares at the top of the lift device in

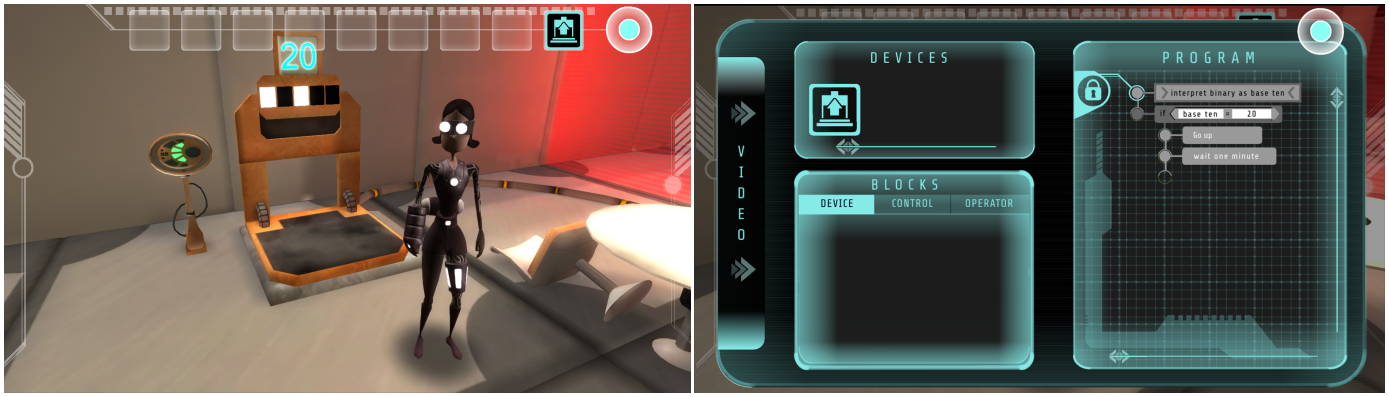

Fig. 2. (Left) A lift device with an existing program in the Digital World level, and (Right) the programming interface displaying the lift's program.

Figure 2, Left) to change the binary sequence until it matches the given base-ten number (Figure 2, Right). Upon executing its program, the lift device evaluates whether the binary sequence equals the base-ten number, and if they match, the device ascends and waits for one minute, which enables the student to navigate a previously inaccessible area. In these tasks, students are provided immediate feedback on the base-ten interpretation of the binary sequence as they flip tiles through a display above the binary sequence (e.g., see numeral '20' in Figure 2, Left).

In the Digital World level, students solve eleven binary representation tasks associated with binary lifts or binary lock devices. The eleven tasks introduce several symbols to represent binary numbers such as "True and False," "Yes and No," and "White and Black," as well as the typical way of denoting them using "1" and "0" to teach the concept of binary representations. Figure 3 illustrates a sample sequence of steps to solve a binary problem for a base-ten number, 20, by a student who is learning the conceptual knowledge about binary numbers. Initially, all binary tiles are off (0), which results in the default base-ten value of 0. If the students flip the fifth (i.e., left-most) bit as in step (a), the base-ten value is updated to 16. Then, if the fourth bit is flipped as in step (b), the value is updated to 24. Then, she notices that the current binary representation makes the value greater than the target value of 20 and decides to flip the fourth bit back to 0 as in step (c). She continues in this manner to find a binary representation that matches 20, executes the program, operates the binary device, and eventually advances to the next task.

It is possible, but not optimal, for the tasks to be solved in a brute-force manner without understanding the concept of binary representations or the programs that control the devices. Therefore, it is critical to dynamically assess students' competency levels in order to provide tailored instructional support for helping students acquire the knowledge. In the following section, we describe the studies we conducted with ENGAGE, which yielded the dataset that we use to investigate deep learning-based stealth assessment.

IV. CLASSROOM STUDIES WITH ENGAGE

ENGAGE was deployed in multiple teacher-led classroom studies conducted in four public middle schools in the southeastern United States. In each round of the study, teachers led a 9-week in-school implementation of ENGAGE. Teachers who led the ENGAGE activities participated in professional development and training sessions before beginning the implementation. Prior to starting the activities students completed pre-surveys (e.g., demographics questionnaire, computer science content knowledge assessment, self-efficacy and computer science attitude surveys). ENGAGE gameplay sessions alternated with classroom activity sessions, and students completed content knowledge tests after completing each ENGAGE game level. Final post-surveys for content knowledge, computer science attitudes, and engagement were administered at the end of the game.

During game-play sessions, ENGAGE was played in either single-player or two-player mode, the latter of which was inspired by prior work on paired programming for introductory computer science [56]. In two-player mode, one student assumed the role of the "driver," who controlled the game using the keyboard and mouse, and the other student assumed the role of the "navigator," who provided guidance and feedback. They collaboratively solving the programming challenges. Students switched roles at pre-defined checkpoints within the game. We posit that paired students shared problem-solving strategies and skills while collaboratively playing the game. Therefore, the same sequence of problem-solving logs was associated with both students in every pair.

A. Assessments and Instruments

Among the assessments and instruments administered during the studies, we utilized (1) a self-efficacy survey, (2) a computer science attitude survey, and (3) a content knowledge assessment. We use students' responses on these instruments

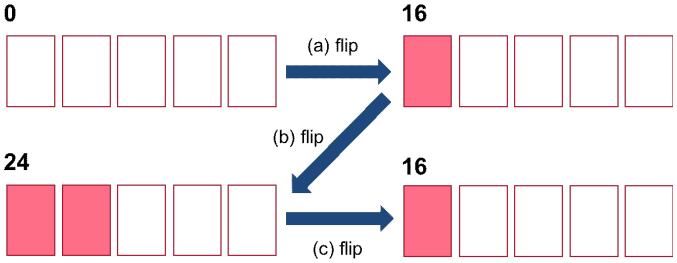

Fig. 3. An example of binary representation learning activities, in which there are three flip actions, step (a), step (b), and step (c).

and their assessment scores to train DEEPSTEALTH's evidence models.

Self-efficacy is the belief in one's capabilities to mobilize the motivation, cognitive resources, and courses of action needed to meet given situational demands [57]. We measure student self-efficacy because previous work has demonstrated that self-efficacy predicts several important work-related outcomes including job performance [58]. Student self-efficacy was measured using the new general self-efficacy (NGSE) scale [28]. Empirical studies suggest that NGSE achieves higher construct validity than Sherer et al.'s general self-efficacy scale (SGSE) [59], while the NGSE scale (8 items) is shorter than SGSE (17 items).

The computer science attitudes (CSA) survey measures attitudes towards computer programming and computer science [29]. The instrument consists of five subscales measuring confidence in learning, usefulness, effective motivation in computer science and programming, attitude towards success in computer science, and attitude on computer science as a male domain. In this work, students completed the three subscales mapping to confidence in learning, perceptions of usefulness, and effective motivation in computer science and programming.

Finally, students completed knowledge assessments developed by the research team to assess how well students mastered concepts in the computational challenges within ENGAGE [60]. We focus on pre- and post-test scores for the items that specifically assess knowledge of computational concepts covered in the Digital World level centering on binary representation. Figure 4 shows example questions in the knowledge assessment.

Q1. How many bits does the binary number 100 have?

1. 3
2. 4
3. 50
4. 100

Q2. Interpret the following binary number as a base-ten number: 10110

1. 3
2. 13
3. 22
4. 10110

Fig. 4. Two sample questions from the concept knowledge assessments.

B. Participants

We analyze interaction data from 191 students (101 males, 88 females, 2 unreported) from a teacher-led deployment of ENGAGE in four public middle school classrooms. Students achieved improvements in content knowledge covered in the Digital World level. A paired *t*-test comparing pre-test (M=0.44, SD=0.21) to posttest (M=0.60, SD=0.25) indicated that students' learning gains were statistically significant with a sizable effect size, $t(184) = 12.18$, $p < .001$, $d = .70$, where 185 out of 191 students took both the pre- and post-knowledge tests.

Of the 191 students, 182 students completed all of the binary representation learning tasks and pre-external learning measures (i.e., NGSE, CSA, knowledge assessment) investigated in this work. Although it is possible to deal with student data with missing values using imputation techniques (e.g., mean imputation) as in [8], [16], we only use data from the 182 students with all valid scores and game interaction logs to minimize any potential noise that might be introduced.

V. DEEPSTEALTH: DEEP LEARNING-BASED STEALTH ASSESSMENT FRAMEWORK

Stealth assessment based on evidence-centered design utilizes three models:

- ***Task Model:*** We use 11 binary tasks from the Digital World level, the objective of which is finding the binary representation that matches the base-ten number specified in an in-game device's program.
- ***Evidence Model:*** Observed sequences of actions in the game reveal evidence of student competencies. A generic feature set is used to represent actions. For ENGAGE, there are 19 possible actions, and thus 19 distinct features are used to represent each action using one-hot encoding, a technique that represents a categorical variable with a binary vector. In addition to the game interaction evidence, students' five pre-learning measures on the knowledge assessment, self-efficacy, and three measures of computer science attitudes are utilized as evidence (i.e., 24 features in total). The evidence model informs the competency model in order to update the stealth assessor's measure of student competencies.
- ***Competency Model:*** We examine one competency model variable with respect to students' overall knowledge about binary representation, where the actual labels for their competency levels are acquired from students' post-test performance on the content knowledge assessment.

As noted above, students interact with 11 binary-lock/lift challenges in ENGAGE, which are defined in the task model. Game interaction logs featuring the series of student behaviors taken to solve these challenges were recorded to a remote MySQL database for post-hoc analyses [61]. Interactions with the tasks reveal action-level evidence about various competencies including one defined in our competency model.

The evidence model processes the raw interaction log data and estimates beliefs about the state of competency variables defined in the competency model. Evidence models generally consists of evidence rules and statistical models [62]. Evidence rules produce observable, predictive features that effectively summarize students' performance from work products, while statistical models, often designed as Bayesian networks, account for estimating beliefs about competency variables given observations.

In prior work, we hand-authored evidence rules to create four features from the raw problem-solving interactions and train deep feedforward neural network (FFNN)-based evidence models [16]. The four key features derived from the evidence rules include the number of binary tile flips, the number of binary tile double flips (i.e., a binary tile flipped and then immediately flipped again), the number of times the device programs were executed, and the amount of time students spent

in the programming interface, which appeared to be important for inferring one's understanding about the concept of binary representations. The features were engineered based on a speculation that students knowledgeable about binary representations were more likely to show fewer binary tile flips, fewer program executions (i.e., they found solutions with fewer attempts), and interpret programs written on the programming interface more quickly. Similarly, for students who gradually learned the concept, they may have exploited double-flip actions to learn the weight associated with each bit in the early phase, but showed fewer double flips as they progressed through and mastered each bit's weight.

While manually engineered features are useful for devising reliable evidence models as demonstrated in [16], feature engineering is a labor-intensive process that requires domain experts' knowledge and substantial effort. Domain experts must scrutinize observable sequences of interactions with given tasks, identify salient characteristics from the observation that could be useful to infer students' levels of proficiency for a set of constructs, and design hand-crafted features that capture the identified characteristics. Further, compared to raw, observed interaction logs, feature engineering often fails to capture fine-grained, sequential information in students' learning behaviors by extracting aggregated, static evidence from low-level trace data.

This work presents the low-level action-based generic feature set that can represent any type of action without being bound to a specific learning environment, thereby yielding enhanced scalability for the stealth assessment framework. In ENGAGE, the binary learning tasks allow 19 possible actions, including 11 pairing actions associated with 11 devices described in the task model (e.g., binary lock device in Figure 2, Left), 5 bit-click actions (e.g., clicking a binary tile in Figure 2, Left), two actions for operating the programming interface (open and close; Figure 2, Right), and a program execution action to run the device's program. Thus, this action-level feature set is composed of 19 low-level features, where each action is represented using one-hot encoding, which is an encoding process that produces a bit vector whose length is the size of the vocabulary of tokens (i.e., 19 actions), where only the associated token bit is on (i.e., 1) while all other bits are off (i.e., 0).

To effectively learn from a sequence of raw action features, we investigate a recurrent neural network (RNN) based evidence modeling approach. RNNs are a type of deep neural network particularly designed for sequence labeling of temporal data. RNNs extract patterns in sequential data and learn predictive features through backpropagation-based training techniques without human interventions. In contrast to FFNNs that assume a fixed length of inputs and outputs, RNNs take variable length sequential inputs while predicting a single output or sequential outputs depending on the task.

Finally, for the competency model, we consider a single competency variable that aggregates students' understanding of binary representations informed by their post-test score on the content knowledge assessment. Each student's data is labeled with a discretized measure of post-test performance that is based on a tertile split (i.e., Low, Medium, or High). Thus, the evidence model's task is cast as a three-class classification problem that infers beliefs about student competency from their raw low-level game trace data and pre-learning measures.

Under this problem formulation, four machine-learning techniques are explored, including two deep learning-based models (deep feedforward neural networks and long short-term memory networks) and two competitive baseline models (linear-chain conditional random fields and naïve Bayes), where every method learns evidence models utilizing low-level action sequences represented with one-hot encoding. Because feedforward neural networks and naïve Bayes classifiers do not support time-series inputs, we adopt *n*-gram encoding that encodes the most recent n actions instead of taking into consideration the entire sequence of actions. Below we describe the two deep learning models utilized in DEEPSTEALTH.

A. Feedforward Neural Networks Pre-Trained Using Stacked Denoising Autoencoders

Deep learning is a family of machine learning techniques grounded in deep artificial neural networks, which are capable of extracting hierarchical representations by inducing multi-level abstractions of training data [27]. Researchers have undertaken a rich line of investigation into how to effectively train deep neural networks (DNNs), including (1) improvements in hardware (e.g., fast CPUs, GPU acceleration, parallel computing), (2) increasing amounts of data including both labeled and unlabeled data, (3) novel neural network architectures along with effective optimization/regularization techniques, and (4) unsupervised pre-training techniques, among others [63]. Deep learning forms the basis for state-of-the-art techniques for a broad range of classification tasks associated with computer vision, speech recognition, and natural language processing [27].

An approach to pre-training DNNs leverages an unsupervised method called autoencoders (AEs), which aim to minimize the reconstruction error of the original input in a DNN without using labels associated with the input [64], [65]. This unsupervised pre-training technique helps to find a region of parameter space that can reach a better local optimum in a non-convex optimization graph, without which optimizing deep neural networks often becomes challenging due to vanishing/exploding gradient issues [66].

More formally, AEs feature (1) encoding (f) that deterministically maps (W_1) an input vector (x) into a hidden representation $f(x)$ using a non-linear transformation characterized by an activation function, s (Equation 1) and (2) decoding (g) that maps (W_2) the hidden representation $f(x)$ back to $g(f(x))$, a reconstructed vector of the input vector (x), using s (Equation 2). The objective in AEs is learning representations (W_1 and W_2) along with two bias terms (b_1 and b_2) by minimizing the reconstruction error between the input x and the reconstructed input $g(f(x))$ through backpropagation methods (e.g., stochastic gradient descent).

$$f(x) = s(W_1 x + b_1) \quad (1)$$

$$g(f(x)) = s(W_2 f(x) + b_2) \quad (2)$$

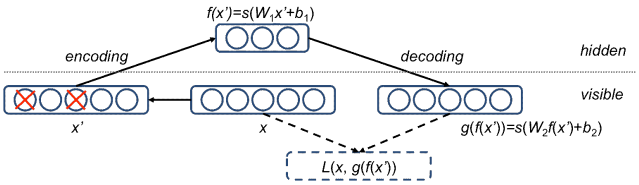

Fig. 5. Illustration of stacked denoising autoencoders; red crosses denote corruption [65].

As a regularized variant of AEs, a denoising autoencoder (DAE) performs a corruption process by injecting noise (we set random units to 0 such as in the dropout mechanism [67]) into the original input vector (x). A DAE aims to recover the original uncorrupted input from the corrupted input as illustrated in Figure 5. In this method, the input vector x is partially corrupted into x' based on the corruption level that defines the probability of corrupting input units. Then, x' is deterministically mapped to $f(x')$ via an encoding process, and $f(x')$ is recovered to the original input x by a decoding process, $g(f(x'))$, by following the standard AE process. A key difference in DAE is that the objective function is to minimize the reconstruction error (L) between the uncorrupted input x and the decoded output based on the corrupted input, $g(f(x'))$, interpreted as denoising corrupted inputs.

We induce feedforward neural network (FFNN)-based stealth assessors pre-trained with stacked denoising autoencoders (SDAEs) [65]. We adopt an approach to training stacked denoising autoencoders using greedy, layer-wise pre-training. Instead of training the deep autoencoders at once, we construct multiple DAEs sequentially from the bottommost layer (i.e., input layer) to the top hidden layer, where previously pre-trained parameters serve to create an input for the next DAE. The objective of each pre-training step is to minimize the reconstruction error of the uncorrupted input. Once the pre-training process is complete, we use pre-trained weight configurations as initial weights for the original network and the entire network gets fine-tuned using the supervised learning criterion. As a result, it has been demonstrated that SDAEs leveraging perturbed input data provide benefits over stacked AEs by effectively dealing with noisy input data utilizing denoising techniques and preventing weights from reaching a trivial solution (i.e., identity function) that could cause overfitting [65].

To fine-tune SDAE-pre-trained models, the input layer is fed with a student's action sequence (the number of actions to consider should be determined prior to training an evidence model) along with the external pre-learning measures, and the output layer is set with the student's competency level.

B. Long Short-Term Memory Networks

Long short-term memory networks (LSTMs) are a variant of recurrent neural networks (RNNs) that are specifically designed for sequence labeling [68]. LSTMs have achieved high predictive performance in various sequence labeling tasks, often outperforming standard RNNs by preserving a longer-term memory and effectively addressing the vanishing gradient problem [69]. LSTMs have achieved state-of-the-art performance in a diverse set of computational sequence-labeling tasks, including speech recognition and machine translation [63].

LSTMs (Figure 6A) feature a sequence of memory blocks. Each memory block includes one self-connected memory cell along with three gating units: an input gate, a forget gate, and an output gate. In LSTMs, the input and output gates modulate the incoming and outgoing signals to the memory cell, and the forget gate controls whether the previous state of the memory cell is preserved or forgotten. The three gating units (input gate, output gate, and forget gate) featured in LSTMs enable modeling long-term dependencies within temporal sequences by allowing gradient information to flow over many time steps.

In an implementation of LSTMs, the input gate (i_t), forget gate (f_t), candidate value of the memory cell ($\tilde{c}_t$), and output gate (o_t) at time t are computed with Equations 3–6, respectively, in which W and U are weight matrices for transforming the input (x_t) at time t and the cell output (h_{t-1}) at time t-1, b is the bias vector of each unit, and σ and *tanh* are the logistic sigmoid and hyperbolic tangent functions, respectively:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (3)$$

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \quad (4)$$

$$\tilde{c_t} = tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (5)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (6)$$

As described in Equation 7, the current memory cell's state (c_t) is calculated by modulating the current memory candidate value ($\tilde{c}_t$) via the input gate (i_t) and the previous memory cell state (c_{t-1}) via the forget gate (f_t). Through this process, a memory cell decides whether to keep or forget the previous memory state and regulates the candidate of the current memory state via the input gate. The current memory cell state (c_t) is controlled by the output gate (o_t) to compute the memory cell output (h_t) of the LSTM block at time t. This step is described in Equation 8:

$$c_t = i_t \tilde{c_t} + f_t c_{t-1} \quad (7)$$

$$h_t = o_t \, tanh(c_t) \quad (8)$$

Lastly, we use the final memory cell output vector (h_t) to predict the class label, which is the belief of the competency level of the student. This step is executed in a softmax layer (top-right in Figure 6A), which is interpreted as a calculation of posterior probabilities of the possible class labels. The LSTM is end-to-end trainable, where all the parameters such as W, U, and b are machine-learned using backpropagation through time.

C. Configuring Deep Neural Networks for Evidence Models

While the output layer of DNN models is fixed to three units (Low, Medium, and High) that represent students' post-test performance (i.e., competency), the input layer size varies according to the model.

Since FFNNs take fixed size inputs, we design an n-gram encoded FFNN architecture to partially capture sequences of actions. N-gram encoding formulates an input using the most recent n actions (i.e., the current action along with [n-1]

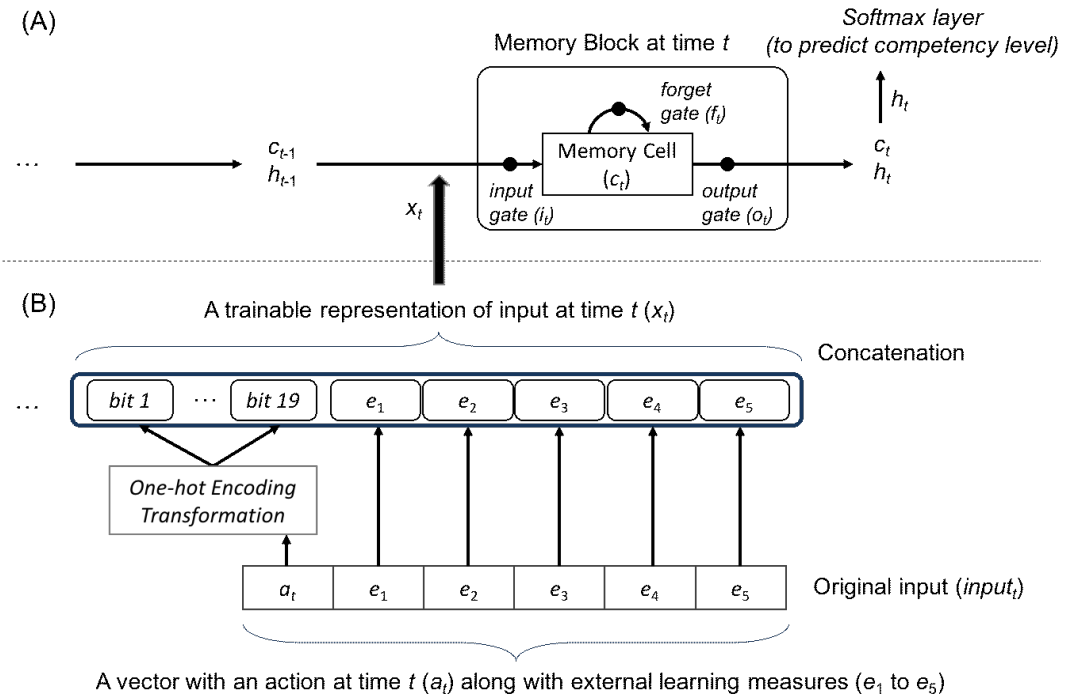

Fig. 6. (A) An illustration of an LSTM memory block that features three gating units and a memory cell [68]. (B) An illustration of how an original input ($input_t$) is transformed to a trainable format (x_t). The discrete action variable, a_t, is one-hot encoded into a 19-dimensional vector using bit 1 to 19, and then the induced vector is concatenated with numeric external learning measure variables (e_1 to e_5) to create the final input, x_t [8].

immediately preceding actions) by concatenating the n actions. In our work, each action is represented in one-hot encoding with 24 features (19 action types + 5 external pre-learning measures), and thus the total number of features is $24 \times n$. In this work, we set n to 200, by which we consider the past 200 actions to predict students' competencies. On the other hand, LSTMs can deal with sequential inputs without constraining the input set to a static size. Thus, as illustrated in Figure 6B, 24 input features are utilized, which consist of 19 action types (*bit 1* to *bit 19*) + 5 external pre-learning measures (e_1 to e_5), and the model extracts temporal patterns from the time-series training data.

As in other machine learning techniques, selecting hyperparameters for deep neural networks often must be empirically determined [70]. We investigate FFNNs with two hidden layers exploring the number of units (256 or 512) per hidden layer. Further, we explore the corruption level (0.5 or 0.75), which is fractional rate of corrupted input units during pre-training, for SDAEs. In a similar fashion, we explore two hyperparameters for LSTMS: the number of hidden units (70 or 140) and the dropout rate (0.5 or 0.75) [67], a regularization technique applicable to neural networks. A grid search method is adopted for each of the DEEPSTEALTH models to perform hyperparameter optimization.

Other than these two hyperparameters, we have fixed the following: (1) for FFNNs, the number of hidden layers is set to two, and Rectified-Linear-Unit and Softmax activation functions are used for hidden layers and the output layer, respectively, and mean squared error and categorical cross entropy are adopted as loss functions for pre-training and fine-tuning, respectively, and (2) for LSTMs, we investigate a single-layer LSTM with the categorical cross entropy as the loss function. For both models, we use the Adam stochastic optimization method [71]. Finally, we set the maximum number of epochs to 100, and model training stops if there is no improvement in the validation accuracy rate within the last seven epochs.

VI. EVALUATION

To answer our three research questions, we conducted an empirical evaluation of DEEPSTEALTH. We first investigate the predictive performance of the four evidence modeling techniques, including two deep learning-based models induced using the DEEPSTEALTH framework. The evaluation was conducted using student-level ten-fold cross-validation, where the student split is fixed across different evidence models to conduct a fair comparison. Then, we explore how each of the sub-feature sets influence the top-performing models' performance during cross-validation, and lastly we report early prediction capacity of the top performing models. Below, we briefly introduce two additional competitive baseline models, conditional random fields and naïve Bayes classifiers. Among the 182 students, 55, 51, and 76 are labeled as Low, Medium, and High performing students based on a tertile split, respectively. Thus, the majority-based baseline accuracy is 41.76%.

A. Baseline Approaches: Conditional Random Fields and Naïve Bayes Classifiers

Conditional random fields (CRFs) are discriminative, undirected graphical models, which are specifically designed to learn interdependencies among output variables for structured

prediction [72]. CRFs are regarded as a sequential extension of logistic regression models or a discriminative analog of hidden Markov models [73]. As a sequence-labeling approach, CRFs have yielded encouraging results in a broad range of structured prediction tasks in natural language processing as well as computer vision and bioinformatics by effectively modeling spatial, contextual relationships characterized in data. We investigate linear-chain CRFs that extract flat, sequential patterns from a series of learning behaviors. Similar to the hyperparameter optimization applied to neural networks, we run a grid search for choosing CRFs' hyperparameters. We investigate two optimization techniques between a one-slack cutting plane method solved using CVXOPT [74] and block-coordinate Frank-Wolfe [75], and the regularization parameter (C) for both optimization techniques among {1.0, 1.5}. The maximum number of iterations over a dataset to find constraints and perform updates is set to 100.

Naïve Bayes classifiers (NBs) are a type of Bayes network that uses a naïve assumption of conditional independence across features. The posterior probability $p(y|X)$ is proportional to the prior, $p(y)$, multiplied by the likelihood of the features, $\prod_i p(x_i|y)$. The distribution of the likelihood, $p(x_i|y)$, such as a Gaussian distribution or Bernoulli distribution, should be determined depending on the characteristics of the data. Our feature set includes (1) pre-learning measures, which are continuous variables, and (2) game interaction logs, which are a categorical variable that are represented in a one-hot encoded binary feature vector. Due to this heterogeneity in the features, we discretize the pre-learning measure features into binary features using a median split obtained from the training set, and then train Bernoulli naïve Bayes classifiers that model both in-game actions and learning features. As in FFNNs, NBs utilize the past 200 actions to predict students' competencies for the current action.

B. Predictive Performance of Evidence Models

Table II shows the results of student-level ten-fold cross-validation results of the four computational evidence modeling approaches using all available features. The rows and columns represent the hyperparameters for each evidence model, and the average predictive accuracy is reported within a corresponding cell associated with a pair of hyperparameters. The performance of the four techniques are evaluated using the same data split per fold for a pair-wise comparison. Each evidence model infers students' competencies derived from their post-test performance utilizing their action sequences and external pre-learning measures.

Results indicate that LSTM-based evidence models (number of hidden units: 140, dropout rate: 0.75, accuracy rate: 63.71, standard deviation: 4.78) outperform the other competitive baselines: FFNNs (number of hidden units: 256, corruption level: 0.5, accuracy rate: 58.80, standard deviation: 5.84), CRFs (one-slack cutting plane method, regularization parameter=1.5, accuracy rate: 61.70, standard deviation: 11.10), and NBs (accuracy rate: 46.63, standard deviation: 14.10) as well as the majority class baseline (accuracy rate: 41.76) in terms of the average competency prediction accuracy. Notably, the LSTM-based models exhibit the lowest standard deviation in test accuracies across ten folds as well as the highest average predictive accuracy.

TABLE II
PREDICTIVE PERFORMANCE: AVERAGE CROSS-VALIDATION ACCURACY RATES OF LSTM, FFNN, CRF, AND NB MODELS

LSTM	**Dropout Rate of 0.50**	**Dropout Rate of 0.75**
70 Hidden Units	57.19	63.22
140 Hidden Units	57.66	**63.71**
FFNN	**Corruption Level of 0.50**	**Corruption Level of 0.75**
256 Hidden Units	**58.80**	57.14
512 Hidden Units	55.53	53.24
CRF	***C* of 1.0**	***C* of 1.5**
One Slack	60.53	**61.70**
Frank Wolfe	60.53	60.53
NB	**46.63**	
Majority Class Baseline	**41.76**	

TABLE III
FEATURE SET-LEVEL ANALYSIS: AVERAGE CROSS-VALIDATION ACCURACY RATES OF THE HIGHEST PERFORMING LSTM, FFNN, CRF, AND NB MODELS

	External Measure Feature	**Game Log Feature**	**Combined Feature**
LSTM	60.50	49.47	**63.71**
FFNN	56.05	50.06	**58.80**
CRF	56.67	52.75	**61.70**
NB	38.00	**46.77**	46.63

C. Feature Set-Level Predictive Performance

To further investigate the features examined in the evidence modeling work, we split the combined feature set into the external pre-learning measure feature set and the game interaction log feature set, and we analyze individual predictive performance of the two sub-feature sets on the best performing LSTM, FFNN, CRF, and NB evidence model architectures presented in Section VI.B. This evaluation is conducted using the same method as in Section VI.B; we use the same student split in ten-fold cross-validation, but we re-train the models utilizing game features or pre-learning measure-based features. As reported in Table III, results demonstrate that the combined feature set yields the highest predictive performance for every model but naïve Bayes, while the external measure feature set yields a higher accuracy rate than the game log feature set for most of the models.

D. Early Prediction Analysis

Since the highest performing evidence models take advantage of the combined feature set, we measure early prediction using all the features except for NB. We adopt *standardized convergence point* (SCP) as a metric to measure models' early prediction capacity [30].

SCP is calculated by $\sum_{i=1}^{m}(k_i/n_i)\ /m$, in which m is the total

number of action sequences, and n_i is the total number of actions in the *i*th action sequence. k_i is contingent on whether predictions on the *i*th action sequence converged to the correct label or not; if converged, k_i is the number of actions after which the stealth assessor consistently makes accurate predictions as in the conventional convergence point metric [76]; otherwise, k_i is $n_i + p_i$, where p_i that is greater than zero is the penalty parameter for the *i*th action sequence. Thus, a lower value is better for this metric.

For example, suppose we have two action sequences (AS_A and AS_B) from two different students (A and B), who demonstrated three and four actions, respectively, and an evidence model's prediction results are as follows:

$$AS_A = Incorrect, Correct, Correct$$
$$AS_B = Incorrect, Incorrect, Incorrect, Incorrect$$

SCP for AS_A is 2/3 since the model consistently makes correct predictions after observing the first two actions, and SCP for AS_B is $(4 + p_B)/4$ since it does not converge to the correct prediction. The penalty parameter (p) should be determined considering the learning environments' characteristics. Our stealth assessment corpus shows that a student's action sequence to complete 11 binary representation learning tasks often takes place in one classroom period (40 minutes). To deal with possible long-term inefficiency driven by learning environments with poor stealth assessment models, we set p_i to n_i, so that every non-converged sequence gets penalized to have SCP of 2.

Table IV shows SCP results for the high-performing evidence models identified in Sections VI.B and VI.C. Using SCP with the aforementioned penalty parameter (lower is better), LSTM shows the best early prediction capacity followed by CRF. SCPs of NBs based on the game log feature set and LSTMs, FFNNs, and CRFs based on the combined feature set are reported.

TABLE IV
EARLY PREDICTION ANALYSIS: AVERAGE CROSS-VALIDATION SCPS OF THE HIGHEST PERFORMING LSTM, FFNN, CRF, AND NB MODELS

	LSTM	FFNN	CRF	NB
SCP	**86.16**	104.77	92.32	122.39

VII. DISCUSSION

DEEPSTEALTH demonstrates significant potential for robust stealth assessment modeling. Addressing RQ1 (overall predictive accuracy), the evaluation reported in Section VI.B indicates that DEEPSTEALTH using long short-term memory networks (LSTMs) (63.7%) outperform three competitive baseline models, including the best performing feedforward neural networks pre-trained with stacked denoising autoencoders (FFNNs) (58.8%), conditional random fields (CRFs) (61.7%), and naïve Bayes models (NBs) (46.6%) in ten-fold cross-validation for predicting student competency on binary representations.

Addressing RQ2 (feature set-level predictive accuracy), the feature set-level analysis (Section VI.C) for the same dataset found that three of the best performing evidence models took advantage of all available features: the game interaction log features and external pre-learning measure features. For LSTMs, a contribution ratio calculated by *a feature set-based model predictive accuracy divided by the combined feature set-based model predictive accuracy* indicates that the external measure feature set and the game log feature set contributes to 94.96% (= 60.50/63.71) and 77.65% (= 49.47/63.71) of the total predictive accuracy, respectively (Table III). The external measure feature set's high contribution ratio inspired us to conduct a correlation test between the pre-test score and the post-test score. A Pearson correlation test indicates that there was a strong, positive correlation between content knowledge pre- and post-test scores, which was statistically significant (r = .702, p < .001). This result suggests why the machine learning methods significantly benefit from the external measure feature set.

The game log feature set yields lower predictive performance than the external measure feature set for the computational evidence models with the exception of NB. However, when the two feature sets are utilized together, the combined feature set further improves the predictive performance compared to models solely leveraging the external measure feature set. In contrast to LSTM, FFNN, and CRF, naïve Bayes could not take advantage of the combined feature set. Overall, NB is not a robust evidence-modeling approach as it achieves low predictive performance across all the three feature sets. In contrast, deep learning models and CRFs show improved performance by utilizing both feature sets over the external measure feature set, where the normalized gain for LSTM, FFNN, and CRF are 8.13%, 6.26%, and 11.61%, respectively. Game interaction logs represent a trajectory of students' progressive learning process, and they provide granular evidence about how students have learned over their prior knowledge. These three models effectively learn from complex patterns between the external learning measures and students' problem-solving behaviors, thereby achieving improved accuracy rates.

Finally, addressing RQ3 (early prediction capacity), because run-time game and curricular adaptation are central objectives of stealth assessment, early prediction (i.e., making consistently correct assessment predictions as early as possible) is an important measure for evidence models. Results (Section VI.D) indicate that LSTM is the most reliable evidence-modeling technique among the set of computational approaches. It achieves the best early prediction score as well as the highest predictive accuracy using the standardized convergence point metric. It is interesting to observe that the rankings for early prediction echo the predictive accuracy results with the best performance yielded by LSTMs followed by CRFs, FFNNs and NBs, while sequence-labeling approaches involving LSTMs and CRFs outperform the FFNNs and NBs assuming a fixed length for inputs.

DEEPSTEALTH demonstrates significant predictive accuracy for stealth assessment, but it is important to note two limitations in the current work. First, the framework is evaluated with evidence models that infer a single competency variable in the ENGAGE game-based learning environment. Evaluating the framework with a multi-task learning capability [77] to deal with a broader range of competency variables (e.g., variables

related to computational thinking practices and computer science concepts [49]) would strengthen the overall reliability of the stealth assessment framework. Second, the deep learning-based framework lacks reliable methods to interpret trained models and explain how assessment decisions are made. More investigation is warranted with respect to model interpretability and explainability to understand deep neural network-based evidence models devised with DEEPSTEALTH.

VIII. CONCLUSION AND FUTURE WORK

We have introduced DEEPSTEALTH, a deep learning-based stealth assessment framework for measuring learners' competency during game-based learning. Adopting a data-driven approach based on multiple weeks of classroom studies within four public middle schools, we formulated three research questions: (1) Do deep learning-based evidence models outperform other competitive approaches with respect to predictive accuracy? (2) Which feature set among game interaction logs, external pre-learning measures, and combined is the strongest predictors? and (3) Which computational model achieves the best early prediction performance? Evaluation results indicate that long short-term memory network-based evidence models outperform three competitive baselines including feedforward neural networks pre-trained with stacked denoising autoencoders, linear-chain conditional random fields, and naïve Bayes models, as well as the majority class baseline, with respect to predictive accuracy and early prediction capacity. A further evaluation of the top three modeling approaches suggests that the highest predictive accuracy is attained when models are devised using all available feature sets by modeling complex, sequential patterns within students' prior knowledge and in-game learning behaviors during interactions with the game-based learning environment. DEEPSTEALTH shows promise for scalability to other learning environments because it directly utilizes low-level action sequences to predict students' competencies. Thus, in contrast to previous work using probabilistic graphical models, evidence models can be easily devised without labor-intensive feature engineering.

These findings point toward three promising directions for future work. First, it will be important to explore other forms of deep neural network-based evidence models for stealth assessment. These include stacked LSTMs and neural models with a self-attention mechanism [78], [79], which may be able to effectively model students' complex learning behaviors for stealth assessment. Alternative fusion approaches handling different sources of the input feature set [80] might further improve the predictive performance of the models as well. Second, in addition to this evidence modeling work, it will be important to investigate competency models that represent fine-grained relationships between knowledge and skills. Finally, it will be important to investigate how game-based learning environments can most effectively leverage deep stealth assessment to support individualized learning experiences, adaptively select learning tasks scaffolding students' problem solving, and support teachers in the classroom.

ACKNOWLEDGMENT

This research was supported by the National Science Foundation under Grants CNS-1138497 and DRL-1640141. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] J. Lester, E. Ha, S. Lee, B. Mott, J. Rowe, and J. Sabourin, "Serious Games Get Smart: Intelligent Game-based Learning Environments," *AI Magazine*, vol. 34, no. 4, pp. 31–46, 2013.

[2] G. T. Jackson and D. S. Mcnamara, "The Motivation and Mastery Cycle Framework: Predicting Long-Term Benefits of Educational Games," in *Game-Based Learning: Theory, Strategies and Performance Outcomes*, 2017, pp. 97–121.

[3] W. L. Johnson, "Serious Use of a Serious Game for Language Learning," *International Journal of Artificial Intelligence in Education*, vol. 20, no. 2, pp. 175–195, 2010.

[4] B. M. McLaren, D. M. Adams, R. E. Mayer, and J. Forlizzi, "A Computer-Based Game that Promotes Mathematics Learning More than a Conventional Approach," *International Journal of Game-Based Learning*, vol. 7, no. 1, pp. 36–56, 2017.

[5] M. Qian and K. R. Clark, "Game-based Learning and 21st Century Skills: a Review of Recent Research," *Computers in Human Behavior*, vol. 63, pp. 50–58, 2016.

[6] M. Kebritchi, A. Hirumi, and H. Bai, "The Effects of Modern Mathematics Computer Games on Mathematics Achievement and Class Motivation," *Computers & education*, vol. 55, no. 2, pp. 427–443, 2010.

[7] G.-J. Hwang, L.-Y. Chiu, and C.-H. Chen, "A Contextual Game-based Learning Approach to Improving Students' Inquiry-based Learning Performance in Social Studies Courses," *Computers & Education*, vol. 81, pp. 13–25, 2015.

[8] W. Min, M. H. Frankosky, B. W. Mott, E. N. Wiebe, K. E. Boyer, and J. C. Lester, "Inducing Stealth Assessors from Game Interaction Data," in *Proc. Artificial Intelligence in Education*, 2017, pp. 212–223.

[9] N. Vannini, S. Enz, M. Sapouna, D. Wolke, S. Watson, S. Woods, K. Dautenhahn, L. Hall, A. Paiva, E. André, and R. Aylett, "'FearNot!': A Computer-based Anti-bullying-programme Designed to Foster Peer Intervention," *European journal of Psychology of Education*, vol. 26, no. 1, pp. 21–44, 2011.

[10] B. C. Nelson, Y. Kim, C. Foshee, and K. Slack, "Visual Signaling in Virtual World-based Assessments: the SAVE Science Project," *Information Sciences*, vol. 264, pp. 32–40, 2014.

[11] N. Alvarez, A. Sanchez-Ruiz, M. Cavazza, M. Shigematsu, and H. Prendinger, "Narrative Balance Management in an Intelligent Biosafety Training Application for Improving User Performance," *International Journal of Artificial Intelligence in Education*, vol. 25, no. 1, pp. 35–59, 2015.

[12] R. Garris, R. Ahlers, and J. Driskell, "Games, Motivation, and Learning: a Research and Practice Model," *Simulation & Gaming*, vol. 33, no. 4, pp. 441–467, 2002.

[13] J. P. Rowe, L. R. Shores, B. W. Mott, and J. C. Lester, "Integrating Learning, Problem Solving, and Engagement in Narrative-Centered Learning Environments," *International Journal of Artificial Intelligence in Education*, vol. 21, no. 1–2, pp. 115–133, 2011.

[14] V. J. Shute, M. Ventura, D. Zapata-rivera, and M. Bauer, "Melding the Power of Serious Games and Embeded Assessment to Monitor and Foster Learning Flow and Grow," *Serious Games: Mechanisms and Effects*, vol. 2, pp. 295–321, 2009.

[15] P. Wang, J. Rowe, W. Min, B. Mott, and J. Lester, "Interactive Narrative Personalization with Deep Reinforcement Learning," in *Proc. International Joint Conference on Artificial Intelligence*, 2017, pp. 3852–3858.

[16] W. Min, M. Frankosky, B. Mott, J. Rowe, E. Wiebe, K. Boyer, and J. Lester, "DeepStealth: Leveraging Deep Learning Models for Stealth Assessment in Game-based Learning Environments," in *Proc. Artificial Intelligence in Education*, 2015, pp. 277–286.

[17] N. Bosch, H. Chen, S. D'Mello, R. Baker, and V. Shute, "Accuracy vs. Availability Heuristic in Multimodal Affect Detection in the Wild," in *Proc. Internatioanl Conference on Multimodal Interaction*, 2015, pp. 267–274.

[18] R. Sawyer, A. Smith, J. Rowe, R. Azevedo, and J. Lester, "Enhancing Student Models in Game-based Learning with Facial Expression Recognition," in *Proc. User Modeling, Adaptation and Personalization*, 2017, pp. 191–201.

[19] W. Min, E. Y. Ha, J. Rowe, B. Mott, and J. Lester, "Deep Learning-Based Goal Recognition in Open-Ended Digital Games," in *Proc. AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2014, pp. 37–43.

[20] V. J. Shute and M. Ventura, *Measuring and Supporting Learning in Games: Stealth Assessment*. Cambridge, MA: The MIT Press, 2013.

[21] L. Rosenheck, C.-Y. Lin, E. Klopfer, and M.-T. Cheng, "Analyzing Gameplay Data to Inform Feedback Loops in the Radix Endeavor," *Computers & Education*, vol 111, pp. 60–73, 2017.

[22] S. Puntambekar and R. Hubscher, "Tools for Scaffolding Students in a Complex Learning Environment: What Have We Gained and What Have We Missed?," *Educational Psychologist*, vol. 40, no. 1, pp. 1–12, 2005.

[23] J. Roschelle, Y. Dimitriadis, and U. Hoppe, "Classroom Orchestration: Synthesis," *Computers & Education*, vol. 69, pp. 523–526, 2013.

[24] I. Tabak, "Synergy: a Complement to Emerging Patterns of Distributed Scaffolding," *The Journal of the Learning Sciences*, vol. 13, no. 3, pp. 305–335, 2004.

[25] R. J. Mislevy, L. S. Steinberg, and R. G. Almond, "Focus Article: On the Structure of Educational Assessments," *Measurement: Interdisciplinary Research and Perspectives*, vol. 1, no. 1, pp. 3–62, 2003.

[26] Y. J. Kim, R. G. Almond, and V. J. Shute, "Applying Evidence-Centered Design for the Development of Game-Based Assessments in Physics Playground," *International Journal of Testing*, vol.16, no. 2, *pp*. 142–163, 2016.

[27] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[28] G. Chen, S. M. Gully, and D. Eden, "Validation of a New General Self-efficacy Scale," *Organizational Research Methods*, vol. 4, no. 1, p. 62–83, 2001.

[29] E. Wiebe, L. Williams, K. Yang, and C. Miller, "Computer Science Attitude Survey," Computer Science, vol. 14, no. 25, pp. 0–86, 2003.

[30] W. Min, A. Baikadi, B. Mott, J. Rowe, B. Liu, E. Ha, and J. Lester, "A Generalized Multidimensional Evaluation Framework for Player Goal Recognition," in *Proc. AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2016, pp. 197–203.

[31] H. Cen, K. Koedinger, and B. Junker, "Learning Factors Analysis – a General Method for Cognitive Model Evaluation and Improvement," in *Proc. Intelligent Tutoring Systems*, 2006, pp. 164–175.

[32] P. I. Pavlik, H. Cen, and K. R. Koedinger, "Performance Factors Analysis – a New Alternative to Knowledge Tracing," in *Proc. Artificial Intelligence in Education*, 2009, pp. 531–538.

[33] S. Sahebi, Y. Huang, and P. Brusilovsky, "Predicting Student Performance in Solving Parameterized Exercises," in *Proc. Intelligent Tutoring Systems*, 2014, pp. 496–503.

[34] W. Min, J. Rowe, B. Mott, and J. Lester, "Personalizing Embedded Assessment Sequences in Narrative-Centered Learning Environments: A Collaborative Filtering Approach," in *Proc. Artificial Intelligence in Education*, 2013, pp. 369–378.

[35] N. Thai-Nghe, L. Drumond, T. Horváth, A. Krohn-Grimberghe, A. Nanopoulos, and L. Schmidt-Thieme, "Factorization techniques for predicting student performance," in *Educational Recommender Systems and Technologies: Practices and Challenges*, 2011, pp. 129–153.

[36] A. Toscher and M. Jahrer, "Collaborative Filtering Applied to Educational Data Mining," *KDD Cup*, 2010.

[37] S. E. Embretson and S. P. Reise, "*Item Response Theory,"* Psychology Press, 2013.

[38] A. T. Corbett and J. R. Anderson, "Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge," *User Modeling and User-adapted Interaction*, vol. 4, no. 4, pp. 253–278, 1994.

[39] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon, "Individualized Bayesian Knowledge Tracing Models," in *Proc. Artificial Intelligence in Education*, 2013, pp. 171–180.

[40] Z. A. Pardos and N. T. Heffernan, "Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing," in *Proc. User Modeling, Adaptation, and Personalization*, 2010, pp. 255–266.

[41] J. I. Lee and E. Brunskill, "The Impact on Individualizing Student Models on Necessary Practice Opportunities," in *Proc. Educational Data Mining*, 2012, pp. 118–125.

[42] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. Guibas, and J. Sohl-Dickstein, "Deep Knowledge Tracing," in *Proc. Advances in Neural Information Processing Systems*, 2015, pp. 505–513.

[43] R. J. Mislevy, J. T. Behrens, K. E. Dicerbo, and R. Levy, "Design and Discovery in Educational Assessment: Evidence-centered Design, Psychometrics, and Educational Data Mining," *Journal of Educational Data Mining*, vol. 4, no. 1, pp. 11–48, 2012.

[44] S. E. Whitely, "Construct Validity: Construct Representation Versus Nomothetic Span," *Psychological Bulletin*, vol. 93, no. 1, pp. 179–197, 1983.

[45] M. H. Falakmasir, J. P. Gonzalez-Brenes, G. J. Gordon, and K. E. DiCerbo, "A Data-Driven Approach for Inferring Student Proficiency from Game Activity Logs," in *Proc. Learning at Scale Conference*, 2016, pp. 341–349.

[46] B. Mott, J. Rowe, W. Min, R. Taylor, and J. Lester, "FLARE: An Open Source Toolkit for Creating Expressive User Interfaces for Serious Games," presented at the *Foundations of Digital Games*, 2014.

[47] "AP CS Principles," http://www.csprinciples.org/. 2016.

[48] S. Grover and R. Pea, "Computational Thinking in K-12: A Review of the State of the Field," *Educational Researcher*, vol. 42, no. 1, pp. 38–43, 2013.

[49] "K–12 Computer Science Framework," https://k12cs.org/. 2018.

[50] J. Maloney, M. Resnick, N. Rusk, B. Silverman, and E. Eastmond, "The Scratch Programming Language and Environment," *ACM Transactions on Computing Education*, vol. 10, no. 4, pp. 16, 2010.

[51] P. S. Buffum, M. Frankosky, K. E. Boyer, E. Wiebe, B. Mott, and J. Lester, "Leveraging Collaboration to Improve Gender Equity in a Game-based Learning Environment for Middle School Computer Science," *Research in Equity and Sustained Participation in Engineering, Computing, and Technology,* pp. 1–8, 2015.

[52] P. S. Buffum, K. E. Boyer, E. N. Wiebe, B. W. Mott, and J. C. Lester, "Mind the Gap: Improving Gender Equity in Game-based Learning Environments with Learning Companions," in *Proc. Artificial Intelligence in Education*, 2015, pp. 64–73.

[53] P. Buffum, M. Frankosky, K. Boyer, E. Wiebe, B. Mott, and J. Lester, "Empowering All Students: Closing the CS Confidence Gap with an In-School Initiative for Middle School Students,", in *Proc. Technical Symposium on Computing Science Education*, 2016, pp. 382–387.

[54] M. Frankosky, J. Creager, E. Wiebe, P. Buffum, K. Boyer, W. Min, B. Mott, and J. Lester, "Game-based Programming Challenges: Stealth Assessment of Student Competencies," *Annual Meeting of the American Education Research Association*, 2016.

[55] B. Akram, W. Min, E. Wiebe, B. Mott, K. Boyer, and J. Lester, "Improving Stealth Assessment in Game-based Learning with LSTM-based Analytics," in *Proc. International Conference on Educational Data Mining*, 2018, pp. 208–218.

[56] N. Nagappan, L. Williams, M. Ferzli, E. Wiebe, K. Yang, C. Miller, and S. Balik, "Improving the CS1 Experience with Pair Programming," *SIGCSE Bulletin*, vol. 35, no. 1, pp. 359–362, 2003.

[57] R. Wood and A. Bandura, "Impact of Conceptions of Ability on Self-regulatory Mechanisms and Complex Decision Making," *Journal of Personality and Social Psychology*, vol. 56, no. 3, pp. 407–415, 1989.

[58] A. D. Stajkovic and F. Luthans, "Self-efficacy and Work-related Performance: a Meta-analysis," *Psychological bulletin*, vol. 124, no. 2, pp. 240–261, 1998.

[59] M. Sherer, J. E. Maddux, B. Mercandante, S. Prentice-Dunn, B. Jacobs, and R. Rogers, "The Self-efficacy Scale: Construction and Validation," *Psychological Reports*, vol. 51, no. 2, pp. 663–671, 1982.

[60] P. S. Buffum, E. V. Lobene, M. H. Frankosky, K. E. Boyer, E. N. Wiebe, and J. C. Lester, "A Practical Guide to Developing and Validating Computer Science Knowledge Assessments with Application to Middle School," in *Proc. Technical Symposium on Computer Science Education*, 2015, pp. 622–627.

[61] G. Zoeller, "Game Development Telemetry," *Game Developers Conference*, 2010.

[62] R. J. Mislevy, R. G. Almond, and J. F. Lukas, "A Brief Introduction to Evidence-centered Design," *ETS Research Report Series*, vol. 2003, no. 1, pp. i-29, 2003.

[63] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[64] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy Layer-Wise Training of Deep Networks," in *Proc. Advances in neural information processing systems*, 2007, pp. 153–160.

[65] P. Vincent, H. Larochelle, and I. Lajoie, "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with

a Local Denoising Criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.
[66] D. Erhan, Y. Bengio, A. Courville, P. Manzagol, P. Vincent, and S. Bengio, "Why Does Unsupervised Pre-training Help Deep Learning?," *Journal of Machine Learning Research*, vol. 11, pp. 625–660, 2010.
[67] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout : A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
[68] S. Hochreiter and J. Schmidhuber, "Long Short-term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[69] A. Graves, "Supervised Sequence Labelling with Recurrent Neural Networks," vol. 385, Springer, 2012.
[70] J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization," *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012.
[71] D. P. Kingma and J. L. Ba, "Adam: a Method for Stochastic Optimization," in *Proc. International Conference on Learning Representations*, vol. 5, 2015.
[72] J. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in *Proc. International Conference on Machine Learning*, 2001, pp. 282–289.
[73] C. Sutton and A. McCallum, "An Introduction to Conditional Random Fields," *Foundations and Trends® in Machine Learning*, vol. 4, no. 4, pp. 267–373, 2012.
[74] T. Joachims, T. Finley, and C. N. J. Yu, "Cutting-plane Training of Structural SVMs," *Machine Learning*, vol. 77, no. 1, pp. 27–59, 2009.
[75] S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher, "Block-Coordinate Frank-Wolfe Optimization for Structural SVMs," in *Proc. International Conference on Machine Learning*, vol. 28, pp. 53–61, 2013.
[76] N. Blaylock and J. Allen, "Corpus-based, Statistical Goal Recognition," in *Proc. International Joint Conference on Artificial Intelligence*, 2003, pp. 1303–1308.
[77] S. Ruder, "An Overview of Multi-task Learning in Deep Neural Networks," *arXiv Prepr. arXiv1706.05098*, 2017.
[78] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv Prepr. arXiv1810.04805*, 2018.
[79] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All You Need," in *Proc. Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.
[80] T. Baltrušaitis, C. Ahuja, and L. P. Morency, "Multimodal Machine Learning: a Survey and Taxonomy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 423–443, 2019.

Wookhee Min received his BS degree in Computer Science from Yonsei University, and MS and PhD in Computer Science from North Carolina State University. He has published 23 conference papers and issued five patents. His research focuses on artificial intelligence centering on user modeling, multimodal analytics, and natural language processing.

Megan Frankosky received her BA in Psychology, BS in Business, and MS and PhD in Human Factors and Applied Cognition from North Carolina State University. Her interests include user experience design, user engagement, mixed methods research, and adaptive and intelligent systems. Currently, she is a Lead User Experience Researcher at Lenovo.

Bradford Mott received his BS, MCS, and PhD in Computer Science from North Carolina State University. He is a Senior Research Scientist at the Center for Educational Informatics at North Carolina State University. His research focuses on game-based learning environments, intelligent tutoring systems, computer games, and computational models of interactive narrative.

Jonathan Rowe is a Research Scientist at the Center for Educational Informatics at North Carolina State University. He received his MS and PhD in Computer Science from North Carolina State University, and his BS from Lafayette College. His research investigates artificial intelligence and human-computer interaction in adaptive learning technologies, with an emphasis on game-based learning environments, multimodal analytics, interactive narrative technologies, educational data mining, and user modeling.

Andy Smith received BS degrees in Electrical Engineering and Computer Science from Duke University and a MCS degree from North Carolina State University. He has published two journal article and ten conference papers. His research interests focus on leveraging machine learning techniques to enhance educational and training technologies.

Eric Wiebe is a Professor of STEM Education at North Carolina State University and a senior research fellow at the Friday Institute for Educational Innovation. His research interests include STEM learning in technology-rich environments, multimodal communication of scientific and technical information, and research-based strategies for helping schools and teachers maximize the potential of new instructional technologies.

Kristy Elizabeth Boyer is an Associate Professor of Computer & Information Science & Engineering at the University of Florida. Her research focuses on dialogue for teaching and learning, including tutorial dialogue and computer-supported collaborative learning. Her work in computer science education spans K-12 and post-secondary contexts through game-based learning and intelligent learning environments.

James Lester is Distinguished University Professor of Computer Science at North Carolina State University, where he is Director of the Center for Educational Informatics. His research on artificial intelligence-driven technologies for education ranges from intelligent game-based learning environments and intelligent tutoring systems to affective computing, computational models of narrative, and natural language tutorial dialogue. He is a Fellow of the Association for the Advancement of Artificial Intelligence (AAAI).