Enriching the Student Model in an Intelligent Tutoring System

Submitted in partial fulfillment of the requirements for the degree
of Doctor of Philosophy
of the
Indian Institute of Technology, Bombay, India
and
Monash University, Australia
by

Ramkumar Rajendran

Supervisors:

Sridhar Iyer (IIT Bombay)
Sahana Murthy (IIT Bombay)
Campbell Wilson (Monash University)
Judithe Sheard (Monash University)

*The course of study for this award was developed jointly by
the Indian Institute of Technology, Bombay and Monash University, Australia
and given academic recognition by each of them.
The programme was administered by The IITB-Monash Research Academy.*

2014

Dedicated

to

my Parents, Teachers and the Almighty

Thesis Approval

The thesis entitled

Enriching the Student Model in an Intelligent Tutoring System

by

Ramkumar Rajendran

(IIT Bombay Roll Number: 08407406,

Monash Student ID Number: 22117954)

is approved for the degree of

Doctor of Philosophy

Examiners

1. ________________ ________________

2. ________________ ________________

3. ________________ ________________

Supervisors

1. ________________ ________________

2. ________________ ________________

3. ________________ ________________

Chairman

1. ________________ ________________

Date: ________________ Place: ________________

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute/the Academy and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Notice 1

Under the Copyright Act 1968, this thesis must be used only under normal conditions of scholarly fair dealing. In particular no results or conclusions should be extracted from it, nor should it be copied or closely paraphrased in whole or in part without the written consent of the author. Proper written acknowledgement should be made for any assistance obtained from this thesis.

Notice 2

I certify that I have made all reasonable efforts to secure copyright permissions for third-party content included in this thesis and have not knowingly added copyright content to my work without the owners permission.

____

Place

____

Signature

____

Date

____

Name

Abstract

An intelligent tutoring system is a computer-based self-learning system which provides personalized learning content to students based on their needs and preferences. The importance of a students' affective component in learning has motivated adaptive ITS to include learners' affective states in their student models. Learner-centered emotions such as frustration, boredom, and confusion are considered in computer learning environments like ITS instead of other basic emotions such as happiness, sadness and fear. In our research we detect and respond to students' frustration while they interact with an ITS. The existing approaches used to identify affective states include human observation, self-reporting, modeling affective states, face-based emotion recognition systems, and analyzing data from physical and physiological sensors. Among these, data-mining approaches and affective state modeling are feasible for the large scale deployment of ITS. Systems using data-mining approaches to detect frustration have reported high accuracy, while systems that detect frustration by modeling affective states not only detect a student's affective state but also the reason for that state. In our approach we combine these approaches. We begin with the theoretical definition of frustration, and operationalize it as a linear regression model by selecting and appropriately combining features from log file data. We respond to students' frustration by displaying messages which motivate students to continue the session instead of getting more frustrated. These messages were created to praise the student's effort, attribute the results to external factors, to show sympathy for failure and to get feedback from the students. The messages were displayed based on the reasons for frustration.

We have implemented our research in Mindspark, which is a mathematics ITS with a large scale deployment, developed by Educational Initiatives, India. The facial observations of students were collected using human observers, in order to build a ground truth database for training and validating the frustration model. We used 932 facial observations data from 27 students to create and validate our frustration model. Our approach shows comparable results to existing data-mining approaches and also with approaches that model the reasons for the students' frustration. Our approach to responding to frustration was implemented in three schools in India. Data from 188 students from the three schools, collected across two weeks was used for our analysis. The number of frustration instances per session after implementing our approach were analyzed. Our approach to responding to frustration reduced the frustration instances statistically significantly–($p < 0.05$)–in Mindspark sessions. We then generalized our theory-driven approach to detect

other affective states. Our generalized theory-driven approach was used to create a boredom model which detects students' boredom while they interact with an ITS. The process shows that our theory-driven approach is generalizable to model not only frustration but also to model other affective states.

Publications from the thesis work

- A Theory-Driven Approach to Predict Frustration in an ITS, *Ramkumar Rajendran, Sridhar Iyer, Sahana Murthy, Campbell Wilson, and Judithe Sheard*, IEEE Transactions on Learning Technologies, Vol 6 (4), pages 378–388, Oct-Dec 2013.

- Responding to Students' Frustration while Learning with an ITS, `To be submitted to the IEEE Transactions on Learning Technologies`.

- Literature Driven Method for Modeling Frustration in an ITS, *Ramkumar Rajendran, Sridhar Iyer, and Sahana Murthy*, International Conference on Advanced Learning Technologies (ICALT), 2012, Rome, Italy.

- Automatic identification of affective states using student log data in ITS, *Ramkumar Rajendran*, Doctoral Consortium in International Conference on Artificial Intelligence in Education (AIED), 2011, Auckland, New Zealand.

Contents

List of Tables

List of Figures

Figure Page

List of Abbreviations

Commonly used abbreviations

CAI	Computer assisted instruction
CBSE	Central Board of Secondary Education
DBN	Dynamic Bayesian Network
DDN	Dynamic Decision Network
FACS	Facial Action Coding System
HMM	Hidden Markov Model
ICAI	Intelligent CAI
ICSE	Indian Certificate of Secondary Education
IGCSE	International General Certificate of Secondary Education
ITS	Intelligent Tutoring System
KNN	k-nearest neighbor
LMT	Logistic Model Tree
MLP	Multilayer Perceptron
NLP	Natural Language Processing
SVM	Support Vector Machine

Abbreviations defined in this thesis

F_i	Frustration Index
RoF	Reason for frustration
SDL	Sub difficulty level
TT	Teacher Topics

Chapter 1

Introduction

E-learning is one of the important fields of research in education. E-learning provides the same learning content to all users irrespective of their age, experience, knowledge and the like. To manage the learning content and to help in teaching process the learning management system (LMS) are used. LMS are used to manage the students' interaction with course content, managing the learning content, and allowing interaction among students and teachers. To provide learning content to the students at anywhere and anytime, Ubiquitous Learning is used. Due to availability of mobile devices like mobile phones and tablet, and high speed internet connection, the learning content can be accessed by the students using their mobile devices, anywhere and at anytime (Ubiquitous learning). However, providing the same learning content to all students can lead to *cognitive mismatch*. Providing difficult learning content to low performers and providing non challenging tasks to high performers is known as "cognitive mismatch". Intelligent Tutoring System (ITS) is used to provide personalized learning content to each student. The focus area of our research work is ITS. Our research focus in e-learning is shown in Figure 1.1*. ITS provides personalized learning content to students based on factors like their performance, prior knowledge, and others [22], [35]. ITS is defined as "computer-based instructional systems with models of instructional content that specify what to teach, and teaching strategies that specify how to teach" [117]. The intelligence in ITS refers to "the adapta-

*The diagram serves to provide general view about our research area and is not the outcome of our research. The diagram does not represent all domains in e-learning. Moreover, the interactions between domains in e-learning may vary.

tion of its tutoring, which means to provide different tutoring to the individual student" [166]. In ITS, the sequencing of the learning content is personalized to avoid a cognitive mismatch. Adapting the learning content based on the student's needs, and personalizing the learning for the student, enables the ITS to work with students of different abilities.

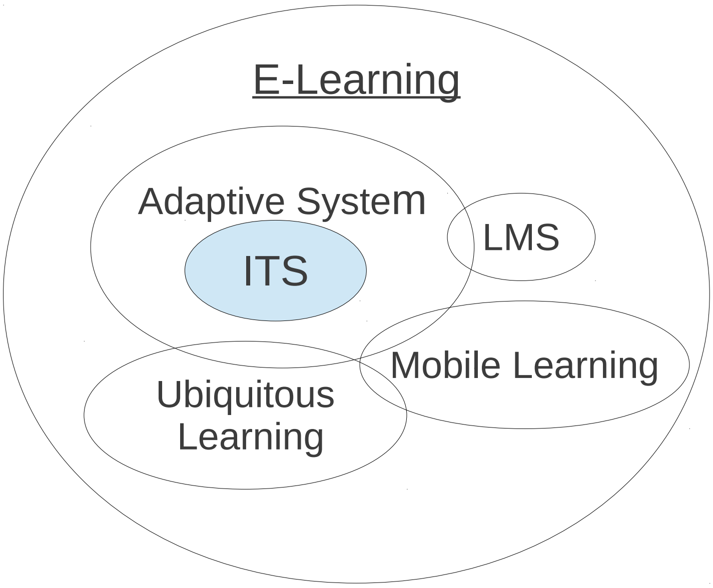


Figure 1.1: Focus Area. LMS = Learning Management System

The generic architecture of an ITS consists of a *student model*, *learning content* and an *adaptation engine*. The student interacts with the ITS by solving problems, or playing games or by other means of interaction. Students' interaction with the ITS, such as responses to questions, number of attempts at a task, and the time taken for various activities (such as responding or reading) are captured in the ITS log file. The log file is analyzed to identify the student's performance and preferences and is stored in the student model. The student model also contains general information about the students, such as class level, previous knowledge and background. The adaptation engine in the ITS tailors the learning content based on the student's request and the data from his/her profile.

To improve the performance of the ITS, affective characteristics of the students should also be considered while tailoring the learning content, as is done in traditional one-on-one learning. D'mello et al.[47] identified the fundamental components of learning as cognition, motivation and emotions. The importance of a student's affective component in learning has led ITS to include learners' affective states in their student models. The consideration of affective processes has been shown to achieve higher learning outcomes in ITS [93], [164].

Recognizing the student's affective states and responding to it is called affective computing [126]. Affective computing was first introduced by Picard [126] and considered in various domains which included gaming, learning, health, entertainment and others [125], [126]. In the past decade, affective computing research generated great interest in affect detection [26]. The affective states widely used in affective computing research in ITS [39], [33], [139], [82], [18], [46] are: frustration, boredom, confusion, delight, surprise and engaged concentration. In our research,we focus on negative affective states like frustration and boredom. We start our research with modeling frustration later we apply our model to detect boredom. To include affective states in the student model, students' affective states should be identified and responded to, while they interact with the ITS. In affective computing, detecting affective states is a challenging, key problem as it involves emotions–which cannot be directly measured; it is the focus of several current research efforts [26], [167].

Methods that have been implemented in ITS to detect the affective state include human observation [39], [134], [133], learners' self-reported data of their affective state [34], [33], mining the system's log file [48], [132], [88], [111], by modeling affective states [33], [139], face-based emotion recognition systems [93], [47], [58], [110], [14], [122], [74], analyzing the data from physical sensors [46], [52], [89], [34], and more recently, sensing devices such as physiological sensors [18], [83], [25], [105]. Advances in these methods to detect affective states look promising in a lab setting. However, they are not yet feasible in a large scale, real-world scenario [133]. The exceptions are data-mining approaches and modeling affective states.

Existing systems which use data-mining approaches [48], [111] have reported high detection accuracy (77% - 88 %) in detecting frustration. On the other hand, the advantage of systems which are based on modeling of affective states is that they not only detect the affective state of the learner but also shed light on the reasons for that state. The existing system [139], [33] which models the affective states reported the less detection accuracy (28% - 70%). Detecting affective states with high accuracy along with the reasons for frustration is still an open research problem. Since the reasons for frustration are not identified in data-mining approaches, the existing system which responds to frustration provides an ad-hoc responses. Hence, an approach to detecting frustration more accurately along with the reasons for the frustration is needed to have an informed responses. This has provided the motivation for us to create a new frustration model–using features constructed from a theoretical definition that will detect frustration with high accuracy along with the reasons for frustration. Hence, our model will respond to the reasons for frustration.

Two responses to affective states in ITS are: a) Adapting the learning content to address the cause of frustration [3] and b) Providing motivational messages to encourage the students to create new goals and to avoid the negative consequences of frustration [93], [133] and [164]. In our research we choose to respond frustration by providing motivational messages.

1.1 Objective

The objective of our research is to create a model to detect and respond to frustration accurately in real-time when students are working with an ITS. To achieve our objective we proposed a theory-based approach. Our proposed methodology is shown in Figure 1.2.

We divide our proposed methodology into four phases:

Phase I: Identify the definition of frustration from educational and cognitive psychology. Identify the methods used to detect frustration from facial expressions. This phase is explained in Chapter 2.

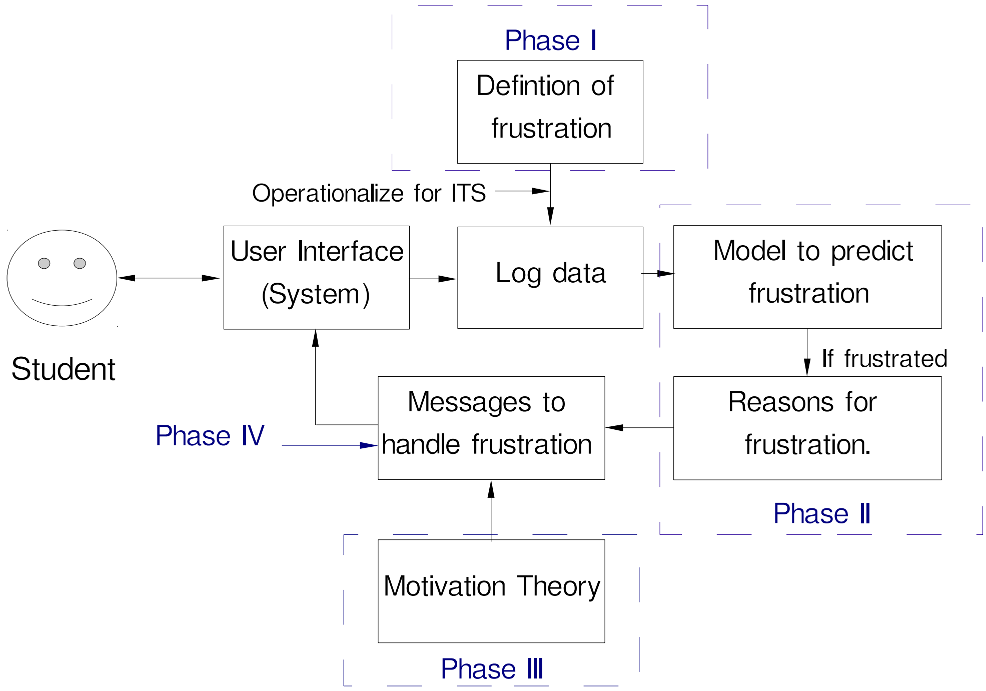


Figure 1.2: Proposed Methodology

Phase II: Operationalize the definition of frustration using the data from the ITS log file. Construct the features to detect frustration using the data from the log file. Create and validate the frustration model using the constructed features. This phase is explained in Chapter 5.

Phase III: Identify the strategies used to respond the negative consequences of frustration. Create the strategies to respond to frustration using motivational messages. This phase is explained in Chapter 2 and 6.

Phase IV: Develop an approach to show motivational messages based on the reasons for frustration. Implement the approach in an ITS and validate its performance. This phase is explained in Chapter 6.

1.2 Solution Approach

To achieve the objectives described in the previous section, we surveyed the literature for definitions of frustration. We considered theoretical and experimental definitions of frustration to define it. We used the definition of frustration from theory as "an emotion caused by interference preventing/blocking one from achieving the goal." We have implemented our research in Mindspark, which is a commercial ITS developed by Educational Initiatives, India, and has been used across schools in India. The log data from Mindspark were analyzed and operationalized, based on the theoretical definitions of frustration. Hence, we constructed the theory-based features from the Mindspark log data. The constructed features are used to create a linear regression model to detect frustration. Human observation to detect frustration was used as an independent method to train and validate the model. We created, validated, and applied an observation protocol based on FACS, and collected the real-time classroom data from 11 students of class six. The dataset was stratified based on the student's Mindspark sessions and a ten-fold cross-validation method was applied to avoid bias in the dataset. The training datasets were used to learn the weights of the frustration model. The gradient descent method was used to calculate the weights of the model. The trained model was tested using the test dataset. The performance measures such as precision, recall and accuracy of the frustration model were then calculated.

To improve the performance of the model, more qualities of frustration, such as the cumulative nature of frustration and time spent to achieve the goal were considered and the frustration model was modified accordingly. Also, we increased our independent dataset to improve the performance. The data was collected in two different schools from two different cities (Mumbai, Ahmedabad) in India to generalize the frustration model for all schools in Mindspark. Cohen's Kappa, ROC analysis, and the F1 score were calculated to validate the performance of the frustration model. The constructed features were applied to a higher order polynomial model and non linear classifiers using Weka–a data-mining software. To compare the performance of existing data-mining approaches on the

Mindspark log data, several features were constructed from the log file without application of theory; the selected features were applied to the different classifiers in Weka. The results of our model are relatively equal to existing data- mining methods in terms of accuracy and relatively better in terms of precision. Also, the reasons for frustration are identified in our model. The performance of the linear regression model is comparatively similar to other models. Hence, for ease of understanding and implementation in Mindspark, we used the linear regression model in our research.

To respond to student's frustration detected by our model, while they work with Mindspark, we integrated the trained model in the Mindspark adaptation logic to detect frustration at real-time. If an instance of frustration is identified by the model, messages are shown to motivate the students to continue the session instead of getting more frustrated. We developed algorithms to show the motivational messages whenever the student is detected as frustrated. The messages are selected based on the reasons for frustration. Our motivational messages are based on a motivation theory–the attribution theory [77], [160], and adheres to the Dweck's [54], [53] theory of praising the effort. The messages were created to praise the student's effort, attribute the results to external factor, to show the sympathy for failure and to get feedback from the students. Our approach to respond to frustration was implemented in Mindspark adaptation logic and tested in real-time. The number of frustration instances per session after implementing the algorithm were analyzed. The frustration instances reduced significantly statistically ($p < 0.05$), due to the motivational messages.

Hence, our objective of detecting frustration accurately along with the reasons for it is achieved; motivational messages to avoid the negative consequences of frustration were implemented and tested in a large scale deployment of the ITS.

1.3 Contribution

In this section, we list the major contributions from our research work.

- We have developed the theory-driven approach to detect affective states using data from the students' interaction with the system. (Chapter 5 and Chapter 7).

- We have developed a linear regression model to detect frustration in a math ITS–Mindspark. The detection accuracy of our model is comparatively equal to existing models to detect frustration and also provides the reasons for why the student is in that state. (Chapter 5).

- We have provided strategies to avoid the negative consequences based on the reasons for frustration. Our approach to respond to frustration significantly reduced the number of frustration instances per session (Chapter 6).

- An approach to detect and respond to frustration is built into a math ITS–Mindspark. Student's frustration instances were detected and motivational messages were provided to avoid frustration in real-time (Chapter 6).

- We found that motivational messages to respond frustration had a relatively high impact on the following kinds of students: (a) those who spent more time in answering the questions; (b) those who had a low average score in Mindspark sessions; (c) those who had a high number of frustration instances per session, before implementing the algorithm (Chapter 6).

- Our theory-driven approach is generalized to detect other affective states. We have developed a model, using theory-driven approach, to detect student's boredom when they interact with an ITS (Chapter 7).

1.4 Thesis structure

This report is organized as:

Chapter 2 discusses the background required for our research work. The functions of Intelligent Tutoring System (ITS), affective states used in affective computing research, the definitions of frustration from educational and cognitive psychology and the motivation theory used in our research to respond frustration are described.

In **Chapter 3**, we briefly describe the Intelligent Tutoring System (ITS) used in our research–Mindspark. Our approach to detect and respond to frustration is implemented in Mindspark and tested in real-time. The details of Mindspark's log data, adaptive logic, learning content and user interface, are described.

Chapter 4 discusses related work on detecting and responding to the affective states of the users. The existing approaches to detect affective states in educational systems like games, Intelligent Tutoring Systems (ITS) and search engines, the process by which frustration is detected and responded to, using data from the students' interaction with the ITS, are described.

Chapter 5 discusses the creation of our frustration model using theoretical definition of frustration. We explain operationalization of our approach using the Mindspark log data. Then we describe our experimental setup which includes human observation, analysis procedure and metrics used to evaluate our frustration model. The results of our model and performance of other existing data-mining approaches applied to the Mindspark log data are discussed. Finally, we describe the performance of different classifier models using our theory-driven constructed features.

Chapter 6 details the messages and algorithm used to avoid the negative consequences of frustration and its performance. **Chapter 7** describes generalizing our theory-driven approach by modeling boredom. **Chapter 8** summarizes this report.

Chapter 2

Background

In this chapter, we discuss the background required for our research work. We start with the functions of an intelligent tutoring system (ITS), and briefly discuss the related research areas in ITS. ITS adapts the learning content based on user preferences and needs. Affective computing is concerned with adapting the learning content based on a user's affective states (emotions), and is one of the research areas in ITS. An overview of the affective states used in affective computing research is provided. In our research we focus on affective computing in ITS, and the affective state considered in our research is frustration. Hence, the definitions of frustration from educational and cognitive psychology are described in detail. Later, we describe the theories and strategies used to respond to affective states.

2.1 Intelligent Tutoring System

An Intelligent Tutoring System (ITS) is a computer system which provides personalized learning content to students based on factors like their performance, prior knowledge, and the like [22], [35]. ITS may be defined as "computer-based instructional systems with models of instructional content that specify what to teach, and teaching strategies that specify how to teach" [117]. The intelligence in ITS refers to "the adaptation of its tutoring, which means to provide different tutoring to the individual student" [166]. In ITS, the sequencing of the learning content is personalized to avoid a cognitive mismatch which may be caused by providing difficult learning content to low performers and providing non challenging tasks to high performers. Adapting the learning content based on the student's needs, and personalizing the learning for the student, enables ITS to work with students of different abilities.

2.1.1 History of ITS

Although Charles Babbage envisioned the programmable computer and created the first mechanical computer in 1800s, the first use of a machine for teaching is reported by Pressey only in 1926 [145]. Pressey used the system to deliver multiple choice questions and to provide feedback immediately. However, this system was not an ITS as it was used only to check whether the response provided by the student was correct or not. In 1960s, programmed instructions (PI) were widely used to teach programming concepts step by step with immediate feedback and remediation for the students [145]. In PI, the problem is solved both by the system and the student. Later, the results are compared and feedback is provided. If the responses do not match, a remedial learning path is suggested. Computer assisted instruction (CAI) or Computer based instruction (CBT) is an extension of PI in a computer. However, CAI is able to provide only predefined learning content based on the student's response. It is not intelligent enough to analyze the student's response and choose the learning content based on the analysis.

Intelligent CAI (ICAI), [161], analyzes the student's response, diagnoses the error and provides remediation. For example, when the sum of $56+47$ is asked, if the student's response is 93, then it indicates that s/he has a problem in the carry over digit in addition, if the student's response is 87, then it indicates that s/he has a problem in addition. Based on the student's response, his/her problem can be diagnosed and then the required learning content can be provided. The ability to diagnose the student's problem and provide remediation is the key difference between ICAI and CAI. The ICAI is also called an intelligent tutoring system (ITS). Hartley and Sleeman in 1973 [76], suggested that ITS must have knowledge of teaching strategies (adaptation strategies), knowledge of the domain (learning model) and the knowledge of the learner (student model). These three features are important features even for modern day ITS. The 'I' (Intelligent) in ITS is the ability to diagnose the student's knowledge, misconceptions, skills and the adapting of the learning content accordingly, rather than providing preprogrammed responses [32], [146] and [145].

At present, ITSs are typically web-based learning systems [23], which are used to teach a wide range of topics from math to medicine [96], [154], [59]. The need for personalized learning and immediate feedback led many researchers to work on the ITS. This resulted in a wide range of ITSs in different fields, for example the SHERLOCK [96] tutor teaches the air-force technicians to diagnose electrical problems in jets, the Cardiac [59] tutor simulates the function of the heart for medical students, Why2-Atlas [154] helps in learning physics principles, and AutoTutor [70] assists students in learning about Internet, operating systems and the like. Math tutors such as Active Math [112], and the Cognitive tutor [131], are widely used in schools as part of the curriculum. In our research work we focus on the web-based ITSs, which are used to teach math in schools. The ITS used for our research, Mindspark, is explained in Chapter 3.

2.1.2 Generic Architecture of ITS

The ITS consists of the user interface, the learning content, the student model, and the adaptation engine [22]. The generic architecture of the ITS is shown in the Figure 2.1. These modules are briefly described below.

User Interface and Log file

The user interface delivers the learning content to the student and accepts the students' responses to the questions posed by the ITS. Based on the nature of the ITS, the learning content can be delivered as text, voice, simulation or interactive games. The user interface can be a mobile device (Tablet, Mobile, Laptop) or a desktop. The students' interaction with the ITS, such as response to questions, number of attempts and time taken for various activities (responding, reading and others) is captured in the log file. The log file

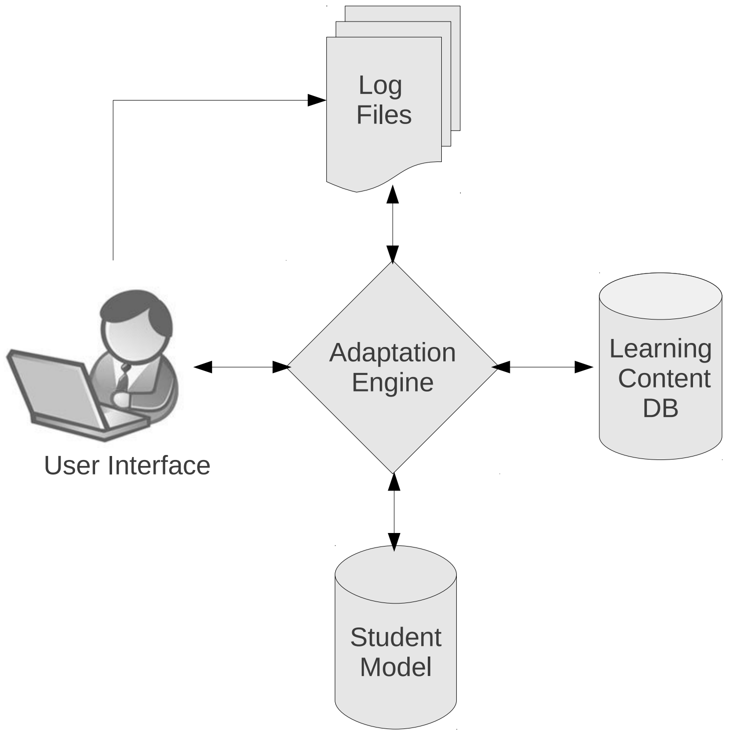


Figure 2.1: Generic Architecture of an Intelligent Tutoring System

is used to create the student model.

Learning Content

The learning content of the ITS is stored in a database as topics. The topics are divided into learning units to enable the teaching of a specific concept or a fact. Misconceptions and common wrong answers of each learning unit are stored in the database. The learning units can be explanations, examples, hints, quizzes, questions and they can be used to teach, demonstrate or test the students.

Student Model

The student model contains general information about the students, such as class level, previous knowledge and background. Apart from this general information, the following information of the students may be stored in the student model.

- Student's skills, goals, plans
- Student's performance such as topic performance and number of questions correctly

answered per session

- Learning characteristics such as the learning rate, the student's preferences and learning styles

- Affective states such as engagement, boredom, frustration and confusion

Adaptation Model

The adaptation engine is an algorithm to select (adapt) the learning content based on the student's input from the user interface (response to the question) and the information from the student model.

ITS adapts the learning content based on the learner's preferences such as:

- The learner's level of ability, such as, "beginner" or "intermediate" or "expert" [102].

- The learner's knowledge, such as, previous knowledge of learning content [80].

- Learning styles such as "visual", "audio", and "interactive" [19, 128].

Function of ITS

In ITS, if a student registers for the first time, general information like age, gender and background are collected and stored in the student model. Based on the student's request, the learning content that matches the student's profile is provided in the user interface. The student then interacts with the ITS by solving problems, or playing games or by other ways of interaction. The student's interaction with the ITS is stored in the log file, which is analyzed to identify the student's performance and preferences and is stored in the student model. The adaption engine in ITS tailors the learning content based on the student's request and the data from his/her profile. For example, if the ITS detects that the student has a misconception on a learning unit, the ITS can suggest that the student redo the current learning unit, or redo the previous learning unit or resolve the misconception using remedial content.

2.1.3 Research areas in ITS

In the past four decades, the research areas in ITS have evolved according to the available technology and user needs. In 1970s the main focus of research was on creating an automatic bug library (misconception and common wrong answers) [24], expert systems [144], tutors (teaching strategies) [31] and a reactive learning environment (system responds to user's actions instead of teaching or correcting the user's mistakes) [20]. In 1980s, along with these research areas, the research focus in ITS was simulations [78], Natural Language Processing (NLP) to improve the student's interaction and introducing case-based learning [143]. In 1990s, the main focus were virtual reality [43], and collaborative learning [90]. In the past decade, the research focus has been on understanding the student's learning instead of teaching strategies and domain experts [21]. The key research areas are the students' cognitive abilities [131], preferences [27], skills [94], learning style [21], learning strategies and affective states [68]. The research focus is on improving the students' learning with the ITS. It is done by predicting a) how students learn [38], b) when students need help [9], c) when to resolve the students' misconceptions [30].

Recent research includes improving the students' interaction with the ITS. Researchers have focused on the user interface to recognize non-textual input [69], interact with students as a human tutor [68], and understand the students' body language and respond to it. Another area of research has been to improve the content of the student model. Students' preferences are identified from their interaction with the ITS. The students' emotions [133], ability and motivation [156] are detected from the log files [132] or by analyzing the data from external devices such as video cameras [93], or from physiological sensors [89]. Researchers have also worked on improving the personalization of the learning content. Students' preferences from the student model are considered while adapting the learning content. Adaptation strategies are researched, in order to enable dynamic adaption of the available learning content for the wide range of student preferences [155].

Researchers have extensively evaluated the students' learning outcomes while learning with the ITS. A recent review article on the impact of the ITS [155], reported that ITSs improve student learning. Most of the commercial ITSs, such as Active Math [112], Cognitive tutor [131], [6], Mathematics Tutor [151], [16], and PAT Algebra Tutor [130], [94] have been tested in schools and have reported an improvement in the students' learning. Moreover, detecting the student's affective states and responding to them (affective computing) has been shown to achieve higher learning outcomes in ITS [93], [164]. In affective computing, the key problem of detecting the affective states is challenging as it involves emotions which cannot be directly measured [26].

2.2 Affective Computing

In traditional learning, affective states, like frustration, may lead the student to quit learning; hence addressing the affective state is important. In a face-to-face teaching situation, experienced teachers observe and recognize the students' affective states and, based on that, change the teaching style to positively influence the learning experience. In ITS, for effective tutoring, students' motivation and affective components should be identified and considered while tailoring the learning content [164], as it is done in traditional one-on-one learning.

Affective computing was first introduced by Picard [126] and considered in various domains including gaming, learning, health, entertainment among others. [125], [126]. In the past decade, affective computing research generated great interest in affect detection [26]. In the next section, we describe the affective states considered in affective computing and the definition of frustration in detail, which is the affective state considered for our research.

2.2.1 Affective States

The typical affective states are emotions, feelings and moods. However, in affective computing research, only emotions are considered [26]. Traditional emotion theories study emotions through facial and body expressions [41], [55], [57], [84], [85]. Emotions were first explored scientifically by Darwin, and Darwin's study reports that some facial and bodily expressions are associated with essential emotions like fear and rejection [41], [26]. Based on Darwin's theory of emotions, six universally recognized emotions have been developed. The six basic emotions are: anger, disgust, fear, joy, sadness and surprise [55], [57], [85].

Emotions have been further researched by cognitive psychologists [147], [142], [135], [120], to emphasize the connection between emotion and cognition [109]. According to cognitive psychology theories such as the appraisal theory, people's perception of their circumstances and interpretation of events determine the emotion, not the event itself. Thus, two people with different appraisals (assessing the outcome of event) or in different circumstances may feel different emotions for the same event [136]. This is the central concept of appraisal theories of emotion [60] and they view appraisal as the cause of the cognitive changes associated with emotions [147].

Later cognitive psychologists [147], [142], [135], [120] explored the cognitive approaches to emotions. This research of cognitive psychologists aimed to understand the relationship between variable (circumstances, goal) and emotion labels (joy, fear) [120] and the relationship between appraisal variables and cognitive responses [99], [147]. The theories of cognitive psychologists state that emotions are related to the student's experience, goal, blockage of goal, achieving of the target, and so on. We discuss two theories of cognitive approaches to affective states in next paragraphs.

Roseman [135] proposed that five appraisals influence emotions: a) motivational state - the motivation is to get a reward or to avoid punishment or failure, b) situational state - whether the motivational state is present or absent, c) probability of achieving the goal, d) legitimacy - deserved outcome of the event, and e) agency - whether the outcome is by self or external person or circumstances. Based on these appraisals the emotions such as joy, pride, distress, and shame are defined.

OCC's appraisal theory [120], explains emotions as a cognitive appraisal of the current situation which involves the events, agents and the objects. It states that, the emotions depend on how one's preferences or goals match with the outcome of that event, that is, the reaction to the events form the emotions. Similarly, the OCC theory discusses the emotions that occur due to the agents and the objects. Based on the outcome of the event, the intervention of the agent, and the objects, the OCC theory defines 22 different emotions such as "Joy", "distress", "hope", "fear" and others. The cognitive emotions (fear, distress) are primarily focused on the student's goals and event outcomes.

Based on research in [44], [101], [45], a review paper on affect detection [26] states that the emotion which occurs during the learning sessions that lasts for 30 to 120 minutes has less relevance with the basic emotions. Hence, the basic emotions might not be relevant to the students' emotions which occur during interaction with computer. Instead learner-centered emotion such as frustration, boredom, confusion, flow, curiosity, and anxiety are more applicable to computer learning environments [26], based on the research in affective computing [39], [33], [139], [82], [18], and [46]. In learner-centered affective states identifying and responding to the negative affective states [95] are important as it might lead the student to drop out of learning. In educational research, there is debate on whether negative affective states like frustration and confusion are required for learning or should be addressed to avoid the students' dropping out [64]. Gee's research on learning through games [64], [65] reports that affective states like frustration can facilitate thinking and learning, hence required while learning. However, Gee [65] further describes that frustration should be kept below certain level to avoid high stress, powerful anger or intense fear. Also frustration is a cause of student disengagement and eventually leads to attrition [88]. Hence, in our research we focus on a negative affective state–frustration. The definitions of frustration are described in the next section.

2.3 Frustration

Frustration has been researched for more than 80 years, hence, classifying or analyzing the various definitions of frustration is beyond the scope of this thesis. Hence we consider the classic frustration theories and experimental definition of frustration in this thesis. We first describe the frustration theories discussed in the book FRUSTRATION: The development of a scientific concept by Lawson in 1965 [97]. The theories considered from Lawson's book are Rosenweig's Frustration Theory [138] and Frustration Aggression Hypothesis by Dollard et al. [50]. The definition of frustration from the textbook "Introduction to Psychology", by Morgan et al. [115], is also discussed in this section. Later we consider the definition of frustration from experimental research by Lazara et al. [98].

2.3.1 Rosenweig's Frustration Theory (1934)

Rosenweig [137] defines frustration as the emotion that an individual feels when a commonly available need or the end-state is not available now, or is removed. For example, a student working in ITS needs help or hint to answer the question, but is not able to get it in the ITS even though help is easily available to him/her in a classroom learning. Rosenweig states that frustration can occur due to the external world or or due to one's own personal actions, such as, the student is not prepared enough to take test. Additionally Rosenweig asserts that "Frustration tolerance tends to increase with age" [138]. We infer that frustration occurs more frequently in school students when compared to college students. Hence, we consider school children in our research.

2.3.2 Frustration Aggression Hypothesis (1939)

The frustration aggression hypothesis [50] states that the existence of frustration always leads to some form of aggression. Frustration is defined as a "condition which exists when a goal-response suffers interference." To understand the definition, we construct an example. This example is used only to explain the definition of frustration aggression hypothesis and not related to ITS. Consider a student prepares for the exams. Based on previous experience, s/he predicts that goal of getting good grades in exams is achievable. His interest to achieve the goal is conveyed using several indicators like not playing games, less time on social networking, and his/her preparation for exam. The strength of these indicators is measured by the duration, force and probability of the occurrence of the goal–getting good grades. In our example, the time spent for preparation (duration), and the probability of getting good score from previous experiences are used to measure the strength of his interest to achieve goal. We only provided the explanation for duration and probability of the occurrence and not force because force is not relevant to measure the strength of indicators in our example. The act which terminates a student's predicted sequence (preparing for exam will lead to good grades in exams) is called goal-response, in our example the students' predicted sequence will be terminated after getting good grades. When the goal-response is interfered with, due to any intervention, such an interference at the proper time in a predicted sequence, is called frustration [50]. In our example, if the student's laptop is being used by someone else like sibling or roommate then s/he gets frustrated. Now the student might scream or hate the person who is using his/her laptop; any such behavior in a goal-response sequence shown towards the interference is called aggression. According to the frustration aggression hypothesis "aggression is the primary and characteristic reaction to frustration".

The hypothesis also states that:

- The greater the strength of the goal-response sequence interfered with, the greater is the frustration; it could affect the strength of the tendency to respond aggressively to frustration.

- The greater the amount of interference with the goal-response, the greater would be the tendency to respond aggressively to frustration.

- The effect of combined frustration can induce stronger aggressive reaction than individual frustration.

In summary, the severity of frustration is determined either by the amount of the interference or by the strength of the interference or the added effect of several frustrations (cumulative).

2.3.3 Frustration and Goal-Blockage

In the classic textbook “Introduction to Psychology”, Morgan et al. [115] define frustration as, “the blocking of behavior directed towards a goal”. The major source of this frustration are environmental factors (a physical obstacle or a person who blocks us achieving our goals), personal factors (not able to attain the goal due to lack of required ability), or conflict (not able to attain the goal due to other goals). This theory also supports Rosenweig’s theory, that frustration can occur due to external or personal factors.

Spector and Paul (1978) [149], further argue that frustration can occur when process of maintaining one’s goal is blocked. Frustration occurs when “both the interference with goal attainment or goal oriented activity and the interference with goal maintenance” [149]. If any goal or expected outcome is blocked or stopped, then a frustration instance occurs. Also, if the process of maintaining the goal is interfered then a frustration instance occurs. The factors that affect the strength of the frustration are the importance of the blocked goal, degree of interference, and number of interferences. The time and the distance to the goal when it is blocked also affects the strength of the frustration [149]. That is, if the goal is missed narrowly then the strength of the frustration is more, when compared to the missing of the goal by a broad margin.

Cognitive psychologists [147], [135], [120], view appraisal as the cause of cognitive emotions occur due to a person’s perspective and expectation of an event. The theories of cognitive psychologists state that emotions are related to the student’s experience, goal, blockage of goal, achieving of the target and so on. We consider the cognitive psychologists’ theory of emotions, that is emotions can occur due to one’s preferences or goal match with the outcome of that event, in our frustration definition.

2.3.4 Experimental Research on Frustration

Apart from these theories of frustration, the following attribute of frustration has been found by experimental research and is not mentioned explicitly in frustration theories.

Lazara et al. [98] studied the causes of frustration in computer users. They concluded that, if a task which is of higher importance to the students involves a large amount of wasted time, then it is directly proportional to a higher level of frustration. It implies that if an important goal, for which the student spent a lot of time fails, it leads to a higher level of frustration. Hence, we should consider the time spent to achieve the goal while detecting frustration.

2.3.5 Definition of Frustration Used in Our Research

From the above research of frustration, the central and primary factor is that frustration occurs when the goal is blocked. Therefore, it is necessary, first, to identify the primary factor of frustration [5], which is, the fact that frustration occurred due to goal-blockage. In addition to the primary factor of frustration, we consider all the characteristics of frustration from the various theories and studies described in Sections 2.3.1 to 2.3.4. However, we do not consider the frustration which occurred due to external interference. Since our focus is to detect frustration that occurs due to a student's interaction with the ITS, we do not consider external factors such as, issues in the device used or connectivity problems.

The following factors of frustration are considered in our research to model the student's frustration.

- Frustration is the blocking of a behavior directed towards a goal [115].
- The distance to the goal is a factor that influences frustration [149].
- Frustration is cumulative in nature [50].
- Time spent to achieve the goal is a factor that influences frustration [98].
- Frustration is considered as a negative emotion, because it interferes with a student's desire to attain a goal [149], [50].

2.4 Motivation Theory

The goal of affective computing is to detect a learner's affective state and respond to it. In the previous sections, we described the theoretical definitions required to detect the affective states. We respond to the student's affective state by displaying motivational messages. Motivational messages motivate the student to continue the task instead of dropping to frustration or goal-failure. In this section, we describe theories behind motivation. Motivation theories are used to motivate the person to get involved in work or to continue working. Motive is derived from the Latin word for movement (movere). Motivation psychologists in 1930 to 1950 studied the factors that moved the resting organism (a living creature) to state of activity. They found that the desire for achievement is the key factor in motivation theory. In 1960s, cognitive motivational theories were developed by researching the application of the motivation theory to the event's outcome (success or failure) [73]. In this section we briefly describe the motivation theories which dominated the scientific study of motivation [73] till 1990. The motivation theories from 1990 are briefed later in this section.

2.4.1 Hull's Drive Theory

Hull's drive theory was developed in the period of 1940-1960 and is based on the energy (drive) required to motivate the organism [81]. Also, it is based on the general law of learning: if stimulus-response (the action towards an event and the response of that event – goal-response) ends with a satisfying result, the motivation increases, and if the stimulus-response ends with an annoying result, the motivation decreases. Hull defines the strength required to increase the motivation, which decreased due to stimulus-response, as habit. In other words, a habit is the action required for a organism to move towards the goal. However, habit can provide the directions required to an action but not the drive. Hence, Hull created the mathematical relation between drive and habit for motivational behavior which is reproduced below:

$$Behaviour = Drive \ \ . \ \ Habit$$

Behaviour is considered as multiplication of “Drive” and “Habit”. This is to indicate that “Drive” or “Habit” alone cannot motivate the organism. If there is no energy (Drive = 0), the organism would not act irrespective of the strength of the Habit. Later, Hull’s drive theory was researched and its results supported the drive theory [10].

2.4.2 Lewin’s Field Theory

Kurt Lewin’s theory was also developed during the period of 1935-1960, just as Hull’s theory. Lewin’s theory is based on Gestalt psychology (the psychology which analyses behavior as a whole and is not determined by the summation of individual elements) to interpret motivational behavior, which is known as the field theory. The field theory states that behavior is determined by both the person and the environment involved:

$Behaviour \;=\; f(Person, environment)$

The motivational force of a person is based on three factors: 1) the person’s intent (need) to complete the task, known as tension (t) 2) The magnitude of the goal (G), which satisfies the need and 3) the psychological distance of the person from the goal (e). The mathematical function for the motivational force of person is: $Force = f(t, G)/e$

In this function the psychological distance from the goal is inversely proportional to the motivation force, that is, if the distance to achieve the goal is reduced (approaching zero), then the motivation to achieve the goal is increased.

2.4.3 Atkinson’s theory of achievement motivation:

Similar to Hull and Lewin, Atkinson also developed the mathematical function for achievement motivation; however, Atkinson focused on individual differences in motivation. Atkinson’s theory states that the behavior (tendency) to approach an achievement-related goal (T_t) is the product of three factors: the need for achievement or motive for success (M_s), the probability that a person will be successful at the task (P_s) and the incentive for the success (I_s). The mathematical function is:

$$T_s = M_s \,.\, P_s \,.\, I_s$$

M_s, the achievement motive, is developed during the early stages of life and shaped by child-reading practices. The probability of success, P_i, usually defined in terms of the difficulty of the task. The value of P_i ranges from 0 to 1. The third factor, I_s, the incentive of success, is inversely related to P_I: $I_s = 1 - P_S$. This theory is developed in the period of 1960-1980.

2.4.4 Rotter's Social Learning Theory:

Rotter's theory is also based on individual differences in behavior like Atkinson's theory. The motivational model by Rotter is based on general expectancy (E) and reinforcement value (RV), and the relationship of these two factors is:

$$Behaviour = f(E, RV)$$

Reinforcement value (RV) is a comparative term and is not clearly mentioned in the theory [73]. The expectancy (E) of success depends on one's history of the present situation and similar circumstances. For example, one's expectancy of success in a event depends on the history of success or failure in the same event or the result of similar events. In a situation which requires one's skill, the expectancy increases after success and decreases after failure. This theory was developed in the period of 1960-1990.

2.4.5 Attribution Theory

Attribution theory considers humans as researchers who are trying to understand the world around them using their own techniques, to reach logical conclusions. Attribution theory, when applied to motivation, considers the person's expectation and the response from the event. Attribution theory was constructed by Heider [77] and subsequently developed by Weiner [160]. Bernard Weiner's attribution theory [160] relates emotional behaviors with academic success and failure (cognitive). In attribution theory, the causes of success and failure related to achievement context are analyzed.

Weiner reports that the reaction of the person is related to the outcome of an event. That is, a person feels happy if the outcome is successful and frustrated or sad if the outcome of the event is a failure. This is termed as "outcome dependent-attribution independent" [160]. The learner's attribution of success or failure are analyzed in three sets of characteristics: locus, stability, and controllability.

- **Locus** refers to the location of the cause, which deals with the cause of success or failure may be internal or external. Locus determines whether the pride and self-esteem are altered due to outcome of an event (success or failure). If the learner attributes the success to internal causes, such as, being well prepared for the exam, and doing more homework, then it will lead to pride and motivates the learner to set new goals. Whereas, if the learner attributes the failure to internal causes then it will diminish the self-esteem. Hence the learner's failure should be attributed to the external factors such as "tough questions" and "difficulty in maths", to motivate the learner to give effort on future event.

- **Stability** refers to the learner's performance in the future. If the learner attributes the success to stable factors such as "low ability", then the outcome of the future event will be the same, given the same environment. If the learner attributes the failure to the stable factors then the future success is unlikely. If the learner attributes the failure to unstable factors such as "less effort" and "luck" then the learner's success in future events will be improved [62].

- **Controllability** refers to the factors which are controllable by the learner and have ability to alter them. If the learner failed the task and s/he can control the future outcome by altering them, such as, improving math-solving ability, spending more time on homework, it will lead to self-motivation. Whereas, if there is failure at a task which s/he can not control, it will lead to shame or anger.

The attribution theory implies that the person's attribution towards the success or failure contributes to the person's effort on future activity. If the learner attributes the success to internal, stable and controllable factors then it will lead to pride, and motivation. If the learner attributes the failure to the internal, stable and non-controllable factors then it will lead to diminishing the self-esteem, shame and anger. Hence, motivating the students' failure by messages which attributes the failure to external or unstable or controllable factors will help them to set a new goal with self-motivation. This theory was developed in the period of 1970-1990.

2.4.6 Discussion

In the above theories, Hull's drive theory and Lewin's Field theory both explain what determines motivation using the same factors: need of a person ("drive" in Hull's and "tension" in Lewin's), the goal object, and directional value ("habit" in Hull's and "psychological distance" in Lewin's). Later, these factors are not considered in expectancy-value theories by Atkinson's and Rotter's and also in the Attribution theory. Atkinson's achievement motivation and Rotter's social learning theory focus on the individual's motivation, success rate, and history. However, these theories addressed the broader goals of motivation and did not provide suggestions to increase classroom performance.

Based on the review article on the above theories by Graham et al. [73], reports that each theory had a life span of around 20 years and major contributions to theories made in those 20 years. The grand theories of Hull, Lewin and Atkinson have not been used after their life span. Also, the research on Rotter's social learning theory has been reduced. Research on the Attribution theory and its application to achievement appears to be dominant in the theory of motivation [73]. Also, the review paper by Graham in 1991 [72], analyzed the papers published in the Journal of Educational Psychology from 1979-1988, related to motivation theories. The study reports that, a) there were 66 published studies in that decade and the primary conceptual framework was the attribution theory, b) "Attribution theory has proved to be a useful conceptual framework for the study of motivation in educational contexts" [72].

More recently (1990 onwards) motivation theory has been researched for its applications. For example, the self-determination theory [63], [127], [42] is an application of motivation theory in organization. This theory discusses the relevance of work motivation in organizational behavior. ExpectancyValue theory of achievement motivation (2000) [162], relates the children's expectancy of success, ability, and subject task to motivation. The recent article [106] in 2004 discusses the future of (work) motivation theory for twenty-first century. The article suggest that the extant theories of work motivations should be integrated, and the work motivation should consider different aspects in organization such as employee motivation, incentives, and performance. The primary focus of research on motivation in the last two decades has been on work motivation applied to organization.

The theory that is still widely used in education research is attribution theory (for example, in [15], and [157]). Attribution theory is also used in affective computing, especially in ITS, to address the students' affective states (for example, in [91], and [45]). Hence, we choose the attribution theory in our research to create motivational messages to address affective states.

2.5 Responding to Frustration

In this section we discuss the different approaches used to respond to frustration in computer based learning environments. In a recent research work by Klein, Moon and Picard [93], listed the strategies to respond to students' affective states. These strategies are developed based on research works on active listening [67] and [118]. The guidelines listed in Klein et.al's strategy in [93] to respond to affective states are discussed below.

- The system should provide option to receive feedback from the student for their affective state. This is to show the student that the system is actively listening to their emotions. Active listening to students' emotions has shown to alter their emotions [118].

- The students' feedback should be requested immediately whenever the student is detected frustrated. The feedback request when the student is not frustrated will be ineffective. To report the affective states, the students' should have list of option to choose from. This will provide the option to student to react on what emotion s/he is undergoing.

- The system should provide feedback messages with empathy, which should make the student feel that s/he is not alone in that affective state. Also the messages should convey the student that the emotion s/he undergoing is valid. For example the student should not feel that only s/he got wrong answers to the question given by the system or only s/he missed the goal.

The other approaches to respond to affective states includes displaying the messages using agents [129], [79]. The agents are designed to show empathy, and encourage the students to continue learning. Also, the research in [123] shows that the positive messages to address the students' emotion have helped them to improve their performance.

To create motivational messages, we studied the Dweck's research [53], [54] on feedback messages to praise the student's effort instead of student's intelligence. In her research, Dweck conducted a nonverbal IQ test on students and provided one of the three forms of feedback messages. One-third of the students were praised for their intelligence, one-third of the students were praised for their effort and remaining students were not praised for effort or intelligence. After providing the feedback message the students were given second set of problems which are difficult compared to first set of problems. Later the students were interviewed to know their view on intelligence. The result shows that the students who were praised for intelligence believes that the intelligence is fixed and cannot be improved. The students who were praised for effort believes that intelligence can be improved by more effort. Also, the students who were praised for effort believes that failure means low effort and displayed more enjoyment in solving difficult problems. The Dweck's research on feedback messages is a seminal work in the research area of guidelines to create feedback messages, and it had been applied to wide range of educational research (examples are motivating school students [163], encourage girl students in maths and science [75], and to respond to students' affective states in computer based learning [44], [39]).

In our research we adapt all the above approaches to respond to frustration. The content in our motivational messages are based on attribution theory [160]. Based on Klein's guidelines [93], we provide the option to students to reflect their feedback, the feedback is requested after detecting frustration and feedback messages to show empathy for students' affective state. Using the recommendation from [129], [79] Our motivational messages are displayed using agents who communicate empathy in their messages. Based on Dweck's research [54], our motivational messages are constructed to praise the effort not their intelligence. Our strategies to respond to frustration is detailed in Chapter 6.

2.6 Summary

In this chapter, we have discussed the function of ITS with its general architecture and recent research areas. After this, we have discussed the affective states in general with a detailed description of frustration. We have discussed frustration from theoretical and empirical points of view, and considered definitions from educational psychology and cognitive psychology. We explained the definition of frustration used in our research and argued how we arrived at the definition. Later, we have discussed five different motivation theories in brief with a detailed description of the Attribution theory which has been used in our research. We also discussed the theoretical guidelines of our approach to respond to frustration. In the next section we describe the functions of the ITS used in our research.

Chapter 3

Research Platform

In this chapter, we describe the Intelligent Tutoring System (ITS) used in our research–Mindspark. Our approach to detect and address frustration is implemented in Mindspark and tested in real-time. The details of Mindspark's log data, adaptive logic, learning content and user interface are described in this chapter.

3.1 Mindspark

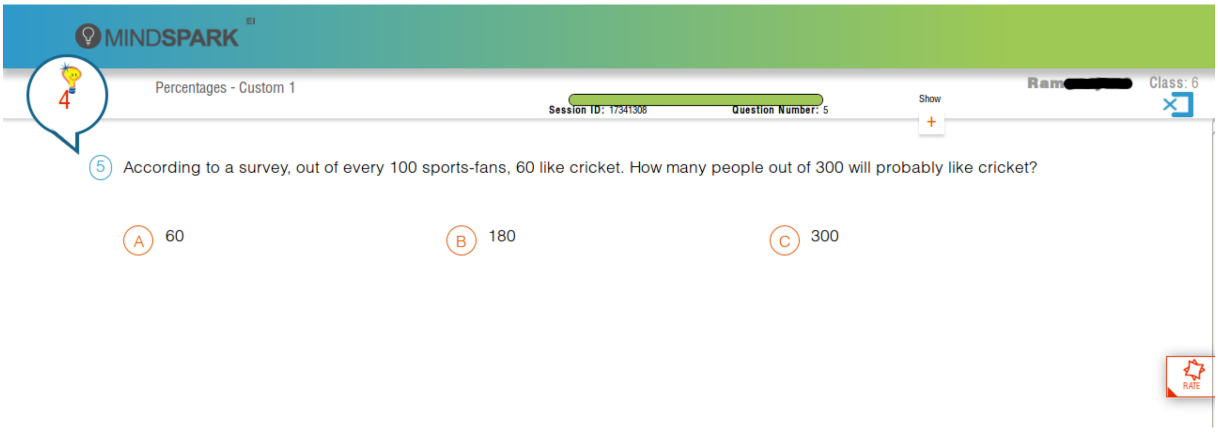
Mindspark is a commercial mathematics ITS developed by Educational Initiatives India (EI-India)[*] [1], and is used in our research to test our proposed approach. Mindspark has been incorporated into the school curriculum for different age groups (grades 1 to 10) of students [150]. Mindspark's learning content is developed to match the school curricula. Schools can select the learning content based on their curriculum like Central Board of Secondary Education (CBSE[†]), Indian Certificate of Secondary Education (ICSE[‡]) or Mindspark Recommended. Also, teacher can customize the learning content for International General Certificate of Secondary Education (IGCSE[§]). Mindspark is currently implemented in more than 100 schools and is being used by 80,000 students at an average of four sessions per week, with each session ranging from 30 to 35 minutes. Mindspark

[*] http://www.ei-india.com/

[†] www.cbse.nic.in/

[‡] www.icseindia.org/

[§] www.cie.org.uk/



(5) According to a survey, out of every 100 sports-fans, 60 like cricket. How many people out of 300 will probably like cricket?

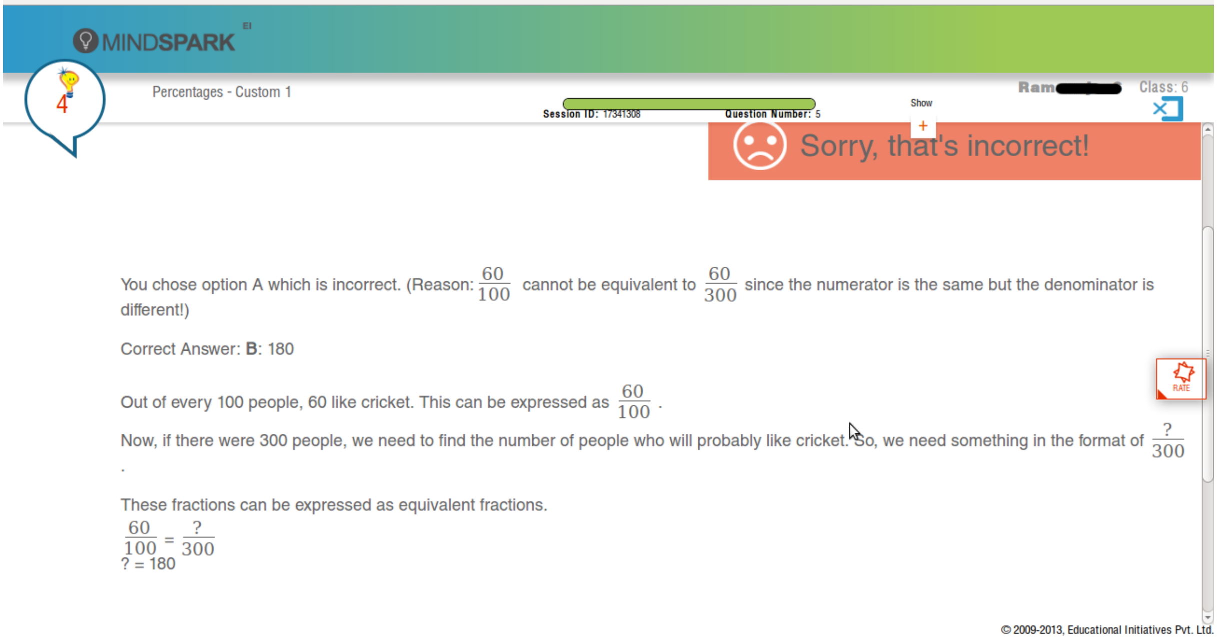
(A) 60 (B) 180 (C) 300

Figure 3.1: Mindspark screen showing a sample question from class 6 on topic Percentages. The Mindspark interface contains student name, class, selected topic, a question from the topic, progress indicator in the topic, number of Sparkies collected in the session, options to change the topic or end the current session, and an option to provide feedback.

is also available as a home-based tutoring system, for a student to work on mathematics problems outside the classroom.

Mindspark is a web-based self-learning system, in which students learn mathematics by answering questions posed by the system. A sample question from Mindspark is shown in Figure 3.1. Mindspark provides detailed feedback and explanations on receiving the answer from the student. A sample explanation from Mindspark is shown in Figure 3.2. The explanation is provided to all questions regardless of the answer provided by the students. Mindspark adaptation logic selects the questions to be presented based on a student's response to the current question and his/her overall performance in the topic, thereby, allowing the student to move at his/her own pace. On the basis of the answer from student, whether right or wrong, Mindspark adapts itself and provides relevant intervention to either strengthen understanding or to offer a higher level of content. If a student performs poorly in the current topic (for example, the student does not answer sufficient number of questions correctly), s/he will be moved to a lower complexity level in the same topic.

Mindspark's architecture is shown in Figure 3.3. Mindspark adapts the learning content from the learning content database. The adaptation is done based on the data from the student model which is created using the student's interaction with the system.



Percentages - Custom 1

Session ID: 17341308 Question Number: 5

Ram[redacted] Class: 6

Sorry, that's incorrect!

You chose option A which is incorrect. (Reason: $\frac{60}{100}$ cannot be equivalent to $\frac{60}{300}$ since the numerator is the same but the denominator is different!)

Correct Answer: **B**: 180

Out of every 100 people, 60 like cricket. This can be expressed as $\frac{60}{100}$.

Now, if there were 300 people, we need to find the number of people who will probably like cricket. So, we need something in the format of $\frac{?}{300}$.

These fractions can be expressed as equivalent fractions.

$\frac{60}{100} = \frac{?}{300}$

? = 180

© 2009-2013, Educational Initiatives Pvt. Ltd.

Figure 3.2: A sample Mindspark screen showing the correct answer with explanation

The student's interactions with the system are stored in log files. In this section we describe the student interface, the log files, the learning content and the adaptive logic of Mindspark.

3.1.1 Student Interface

Mindspark is a web-based learning system which can be accessed using Internet browsers like Mozilla Firefox or Google Chrome. Hence, Mindspark can be accessed using a laptop, a desktop computer or a tablet. In Mindspark, after successful registration, each student gets a login ID. The student logs in with his/her unique ID and password. The teacher assigns the topics to the students based on their curriculum.

The user interface, as shown in Figure 3.1, contains a progress indicator at the top the screen. The progress indicator shows the student's progress in the selected topic. The student's name, class and topic name are displayed above the question. The students have the option of changing the topic or ending the session at anytime during their interaction with Mindspark. A optional emote tool bar is integrated in Mindspark interface (bottom left in the Figure 3.1 indicated as Rate) to enable the students to provide feedback. The

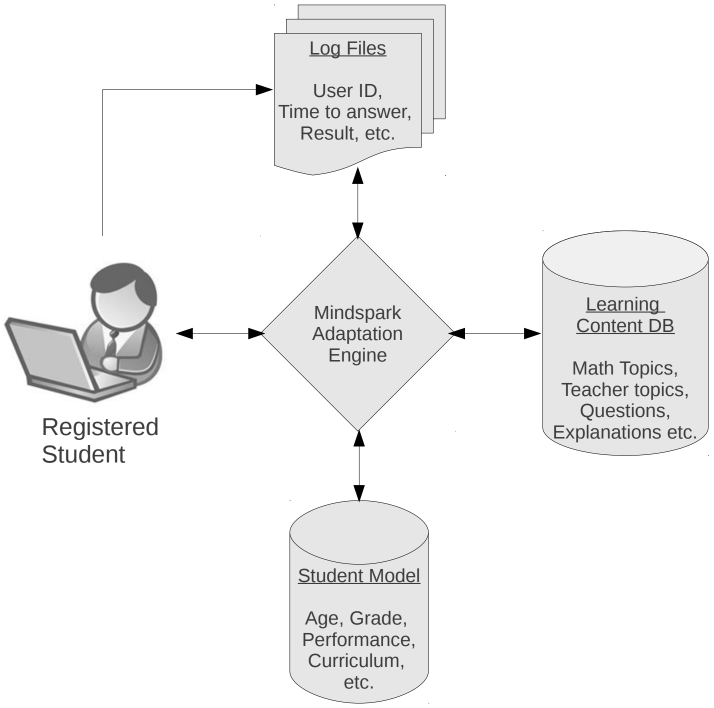
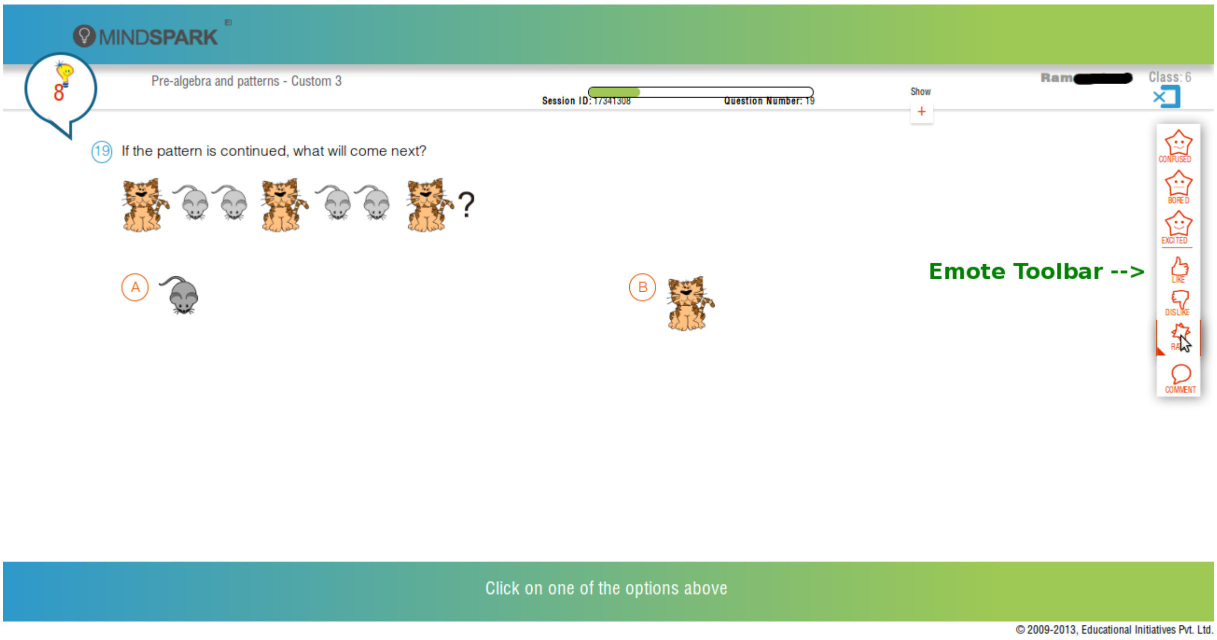


Figure 3.3: Mindspark Architecture - Intelligent Tutoring System

options in the emote toolbar are shown in Figure 3.4. The emote toolbar is used to collect the student's emotion during the Mindspark session. The options in the emote toolbar are "confused", "bored", "excited", "like", "dislike" and an option to receive the student's comments.

Apart from the emote toolbar, based on the student's response to the question, students are allowed to provide feedback on questions, as shown in Figure 3.5. The student can mark whether the question is a repeat question or not. If the student's response to the question is incorrect then student can report whether Mindspark marked his/her response as correct or not. Regardless of the answer provided by the student, the explanation of the correct answer will be provided in the Mindspark interface after submitting the answer. The sample explanation is shown in Figure 3.5.

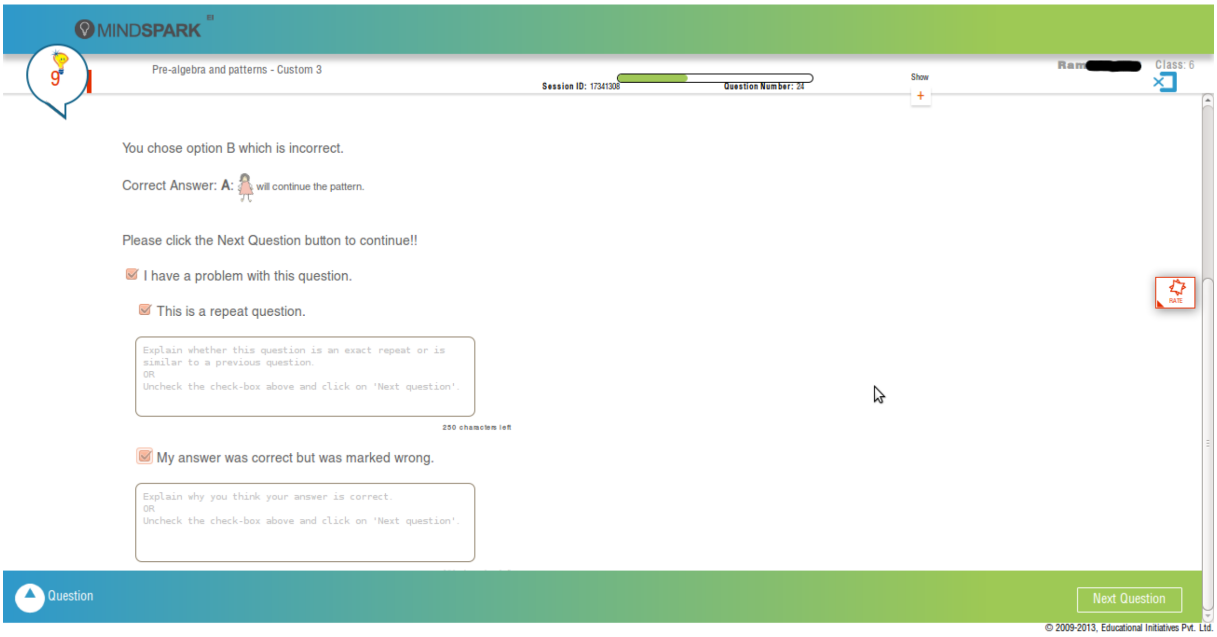
Similar to the student' interface, Mindspark has a teacher interface for school teachers. Teachers can select and a assign the topic for the students based on his/her class curriculum. Also, the teacher can track: a) each student's progress session-wise or topic-



19 If the pattern is continued, what will come next?

Click on one of the options above

Figure 3.4: The options in emote toolbar. The options are “Confused”, “Bored”, “Excited”, “Like”, “Dislike” and “comment” to receive the student’s feedback



You chose option B which is incorrect.

Correct Answer: A: will continue the pattern.

Please click the Next Question button to continue!!

☑ I have a problem with this question.

☑ This is a repeat question.

Explain whether this question is an exact repeat or is similar to a previous question.
OR
Uncheck the check-box above and click on 'Next question'.

250 characters left

☑ My answer was correct but was marked wrong.

Explain why you think your answer is correct.
OR
Uncheck the check-box above and click on 'Next question'.

Figure 3.5: The options for the student’s to enter their feedback when the question s/he answered is incorrect

wise, b) the concept that elicits a low performance from the class, or c) the misconceptions of the students.

3.1.2 Learning Content Database

Mindspark's learning content is stored in the learning content database and are hierarchically arranged in the following order:
Topics $->$ Teacher Topics $->$ Clusters $->$ Sub difficulty Level

- Mathematics is divided into several topics. The topics in Mindspark are: Algebra, Fractions and Decimals, Geometry, Measurement, Mensuration, Number Theory, Numbers, Percentages & Commercial Math, Problem Solving, Ratio-Proportion, Real Numbers, Statistics & Data Analysis and Trigonometry.

- Each topic is divided into Teacher Topics (TT). TT contains subtopics in each topic. For example, the subtopics of "Real numbers" are: Squares and square roots, Logarithms, Cubes and cube roots, Integers, Rational numbers, Real numbers and Exponents and roots.

- Each TT is divided into clusters (learning units). A cluster contains the learning content in the form of questions. Each cluster has 20 to 40 questions. The questions are arranged in progressively increasing levels of complexity.

- The questions in the clusters are divided into sub difficulty levels (SDL). For example, if the cluster is "Squares" the first SDL is " the concept of the square", the second SDL is "Square of a small number and single digit numbers" and so on. Each SDL contains three questions with the same level of difficulty.

In Mindspark, questions are designed to identify the causes of error based on the options students select. The options provided to the students in each Mindspark question is created to diagnose the student's misconception, lack of concept understanding or the common error in that question. The student's misconceptions are addressed using special learning content called a "remedial cluster". The remedial cluster contains examples, animated displays, and interactive multimedia to explain the concept and to resolve the misconception. Apart from the questions and answers, Mindspark's learning database contains feedback with explanations to each question. Explanations are targeted at specific mistakes which are diagnosed from the response provided by the student.

Mindspark's learning content is designed to integrate with the curriculum of a school. So, Mindspark can be configured to match the curriculum each school uses. Mindspark allows teachers to customize the topic according to how the topic is being taught in the school. Teachers can select standard curricula like CBSE, ICSE, Mindspark Recommended or can customize for IGCSE as per the school curriculum. The teachers can then select the topic and subtopic to be activated and customize it according to their requirements. Teachers can view the sample questions from each subtopic before activating it. The activated topic can be assigned as homework or revision, based on the students' performance.

3.1.3 Browser Log and Student Model

The browser log file records the students' interactions with Mindspark. The following details are recorded in the log file: TT (Teacher Topic) attempt ID, user ID, TT code, TT attempt number, attempt time, class level result (values: success, failure, aborted or null), individual result, cluster ID, failed clusters, last modified date, TT cluster status, cluster attempt ID, cluster code, cluster attempt number, result, start session ID, end session ID, serial number, attempted date, question number, answer, time in seconds to answer the question, result (correct or wrong), and time taken to read the explanation. The log files are stored as tables in a cloud-based server. The database is archived at the end of school year (May month).

The student information is captured from the browser log and stored in a separate database called the student model. The student model contains the student's performance in each topic, grade, school curriculum, the student's overall progress and the personal details of the students. Using the student model and data stored in the database, teachers can see the students' learning patterns in the teacher interface. The teacher can have detailed insights into the students' understanding and their misconceptions, right after the topic is taught in the class. It allows the teachers to address the learning gaps of the class. The teacher can learn exactly which concept a student is struggling with, allowing them to offer personalized intervention.

3.1.4 Adaptation Engine

Mindspark delivers the learning content based on the student's performance in the current question, performance in the current topic and the student's grade. In Mindspark, adaptation happens at two levels–the cluster level and the inside cluster level. This Mindspark adaptation logic described in this section is based on 2011 Jan. The adaptation logic of Mindspark is refined later.

3.1.4.1 Cluster level adaptation logic

The students' performance are monitored in Mindspark by analyzing the log files. Progress in each topic is measured and used to determine whether the student has learned the subtopic or not. The adaption logic at the cluster level is given below:

- When a student gets 25% or more questions incorrect in a cluster, it is considered as a cluster failure. On the first failure of the cluster, Mindspark allows the student to repeat the cluster.

- On the second failure of the cluster, the cause of the student's failure is identified based on student's performance in the cluster. If the student gave the wrong answers due to lack of understanding of fundamental concepts, adaptive logic takes the student to the cluster that was immediately before the current one. If it is the first cluster in the topic, the student is taken to a lower level of the same topic. If the student gave the wrong answers due to misconception, then the remedial cluster is given to the student. Remedial cluster is a cluster which explains the concept with more hints and explanation. This cluster teaches the concept of using questions like other clusters. Remedial clusters are designed to resolve the misconception based on cognitive dissonance theory [19]. In the cognitive dissonance method, the system conflicts the student's prior understanding (misconception), then gives the correct explanation to resolve the misconception.

- On the third failure, the student is given five clusters prior to the current cluster, irrespective of the position of the cluster and the class level.
- After the fourth failure, the cluster is highlighted in the teacher interface. The student is allowed to proceed to the next cluster. The teacher can help the student to understand the concept and clear any misconceptions.
- Success on a cluster wipes out past failures.

If a student on the remedial cluster fails twice, irrespective of whether the remedial cluster is from the same TT or some other TT, it will take the student to one cluster lower in the flow. The logic of the third and the fourth failure does not apply to the remedial cluster.

3.1.4.2 Adaptation within a cluster

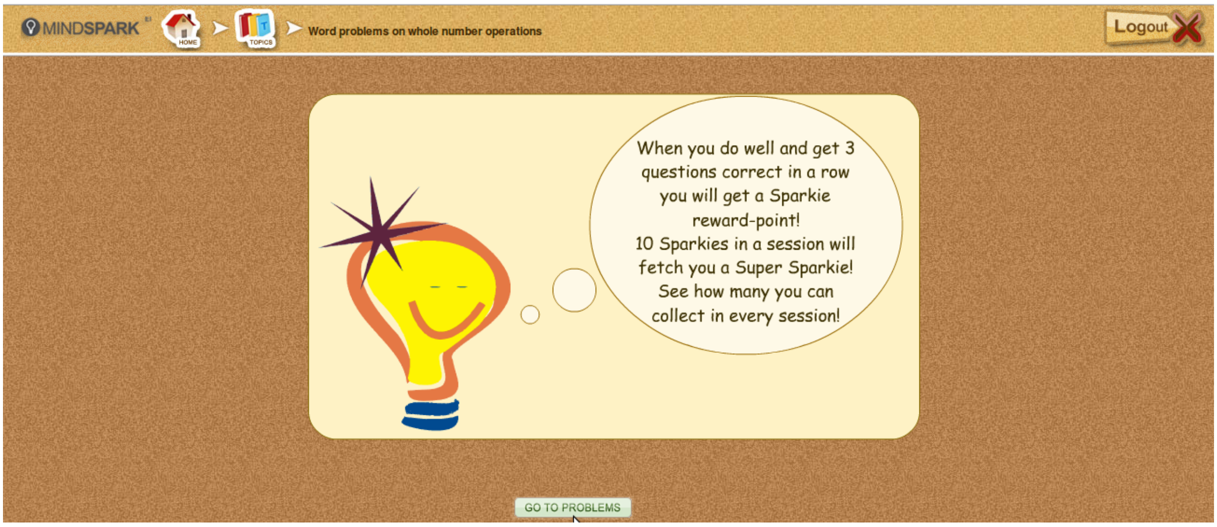
In Mindspark, when a student answers a question incorrectly, the system provides a detailed explanation to the student, on the concept involved. The student is then presented another question of the same difficulty level (DL) from the same learning unit, to give the student another opportunity to demonstrate his/her understanding. The adaptation logic is given below:

- If a student fails to answer one question in the SDL, the next question in the same SDL is given. After attempting three questions, the student is moved to the next SDL, even if the student fails to answer all questions in the SDL.
- If students attempted 50% of questions in the cluster and the total mark percentage is less than 60% in the 50% of the questions attempted then it is considered as a cluster failure. Mindspark will not allow the student to complete the remaining questions.

Mindspark's adaptive logic diagnoses the learning gaps in each student. Then Mindspark helps the student to clear the gaps by providing the required learning content. Also the adaptive logic allows the students to learn beyond their grade level.

3.1.5 Sparkies

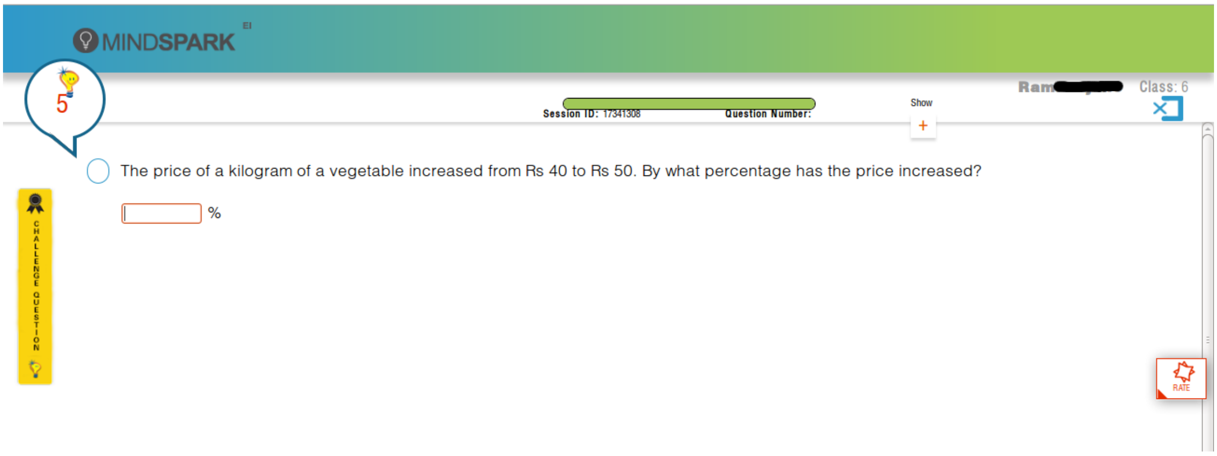
Mindspark rewards the students for their performance. Sparkies are the reward points in Mindspark and they are displayed in the student interface. The Sparkie is introduced to the student when s/he logs into Mindspark and the rules of earning Sparkies are described as shown in Figure 3.6.



When you do well and get 3 questions correct in a row you will get a Sparkie reward-point!
10 Sparkies in a session will fetch you a Super Sparkie!
See how many you can collect in every session!

Figure 3.6: Sparkie is introduced to the students and the rules of earning Sparkies are described

Sparkies act as motivational factors for the students in Mindspark. If the student answers three questions consecutively correct, s/he receives a Sparkie (extra motivational points). If the student answers five questions consecutively correct, s/he receives a challenging question which is tougher than normal questions and can be from any topic that the student has covered. The sample challenge question from topic percentages is shown in Figure 3.7. If the student answers the challenging question correctly, s/he receives five Sparkies. The Mindspark interface in the figures above, show the number of Sparkies the student collected in that session. Every week, students with highest number of Sparkies are identified and their names are published on the Mindspark's login page [¶].



○ The price of a kilogram of a vegetable increased from Rs 40 to Rs 50. By what percentage has the price increased?

□ %

Figure 3.7: Challenge Question from Topic Percentages

3.2 Discussion

The features of Mindspark, such as, feedback with explanation, adapting questions based on the students response and logging of the student's response, time taken to answer and other common data are available in most of ITS. Also Mindspark is used as part of regular school curriculum, hence:

1. It has data of more than 80,000 students

2. It is an opportunity to implement our approach and test it in the real world.

Due to these reasons we chose Mindspark as our research platform. In this chapter, we have described the functions of the research platform used in our research.

[¶] http://www.mindspark.in/login

Chapter 4

Related Work of Affective Computing

In this chapter we review related work on detecting and responding to the affective states of the users. First, we start with existing approaches to detect affective states in educational systems like games, Intelligent Tutoring Systems (ITS) and search engines. Then we discuss in detail, the process by which frustration is detected and responded using data from the students' interaction with the ITS. We conclude the chapter with the motivations for our solution approach.

4.1 Affective States Detection

Adapting the learning content based on the user's affective states is called "affective computing." In affective computing, systems detect and respond to users' affective states [126]. Detecting the affective states has been the focus of the affective computing research area for the past decade [26]. In this section, we first discuss the different approaches used to detect the affective states in various systems such as games, ITS and search engines. Then we discuss existing research to detect frustration using the students' interaction with ITS or educational games.

The students' affective states can be detected by analyzing the data from:

- Facial expressions.
- Body language and postures.

- The physiological sensors.

- Voice or text input.

- Interactions with the system.

The different approaches for affect detection described in this section is based on the recent (2010) review paper on affect detection [26]. This paper discusses five studies using facial expressions, two studies using body language and postures, twelve studies using data from the physiological signals and sixteen studies using voice and text input. Apart from the approaches given in the review paper [26], we included the approach to detect affect states using student's interaction with the system. We also included the research papers published from the year 2010-2013 for each approach. We conclude this section with comparative analysis of different approaches and applicability to our research work.

Facial Expressions

Affective states can be inferred from the users' facial expressions during their interaction with the system (ITS, Games, and the like). The users' facial expressions during their interaction with the system are captured using a video camera or web camera. Later the facial expressions are coded as action units (expressions in different parts of our face, for example, inner brow rise, nose wrinkle, chin rise, jaw drop, and dimpling) [14]. Based on Ekman's Facial Action Coding System (FACS) [57], emotions are identified from the action units [110]. However, manual coding of facial expressions is a time-consuming process [51], and cannot be used in real-time. Hence, automated facial coding systems are used in affective computing. The videos of facial expressions are analyzed automatically and the affective states are detected [58]. In a research paper [58], the mental states like "agreeing", "concentration", "disagreeing", "interested' ", "unsure", and "thinking" are detected using the head movements and facial expressions. The facial expressions considered to detect the mental states are "eyebrow rise", "lip pull", "lip pucker", "lips part", "jaw drop", "teeth." Dynamic bayesian networks (DBN) was used to model the head movement and facial expressions to mental states. This approach reports an overall detection accuracy of 77.4 % [58]. The research study in [14] and [153] classifies facial action units instead of emotions. The facial actions units like "lip pull", "eyebrow rise" are classified using non-linear classifiers, like the support vector machine (SVM), Adaboost and the hidden Markov model (HMM); these approaches result in higher detection accuracy rates–over 90 %. The facial action units are used to classify the basic emotions such as "anger", "fear", and "happy."

Body Postures

Similar to the facial expressions, body language and postures are also used to detect the affective states. The research in [36] reports the correlation of emotions like anger and fear with body movements, body rotations in angles like chest, abdomen, head bend, shoulder and others. The automated approaches use the signal from pressure sensitive chairs (pressure pads on the chair) to identify the user's positions of sitting like leaning forward or back, sitting upright, or sitting on the edge of the seat. These positions of sitting are used to classify the user's interest level (high or low) using the HMM classifier [116] or the user's emotions using the non-linear classifiers [49]. These approaches report detection accuracy in the range of 71 % to 80 %. The users' postures and body motion are analyzed automatically to detect the users' engagement with game companion (robot) [141]. The body motions like "body lean angle" and "contraction index (degree of contraction and expansion of the upper body)" are considered to detect the user's engagement. The classifier in Weka [108] are used in this research and this approach reports 82% detection accuracy.

Physiological Sensors

Affective states can be detected by analyzing data from the physical sensors such as the blue eye camera [46], posture analysis seat [8], the pressure sensitive mouse and skin conductance bracelet [164]. The data from the sensors are compared with the affective states identified using independent methods to detect the affective states. In Wayang Outpost, an adaptive tutoring system for geometry [8], a linear regression algorithm was used to detect affective states using the data from sensors. This study reports the correlation coefficient of multiple features, R = 0.72 for frustration. More recently, the data from physiological sensors such as electrocardiogram (ECG) [18], facial electromyography (EMG) [83], galvanic skin response (GSR) [18], electrodermal [92] and respiration changes [159] are being considered in affective computing. The data from sensors are modeled using classifiers, to detect the affective states. The classifiers used in this approach are k-nearest neighbor (kNN), linear discriminant function (LDF) [159], SVM and multilayer perceptron (MLP). These approaches report the detection accuracy as being in the range of 80-92 % [159], [105] and [17].

Voice and Text Input

The user's input to the system, such as voice and text data, are used to detect their affective states [167]. The following features are considered in voice analysis:

- Voice level, voice pitch and phrase.
- Acoustic-Prosodic features like the speaking rate, the time taken for each speech turn and the amount of silence in each turn.
- Lexical items in a user's response or the system's response.

These voice features are used to classify generic emotions such as anger, fear [100], or to classify emotions at an abstract level such as positive, negative, emotion, non-emotion [104].

Similarly, from the user's text input, features like first sentence, direct speech in sentence, special punctuation (! and ?), complete upper-case word and sentence length in words are used to classify the users' emotions. Classifiers used in this approach are linear classifier [4], Naive Bayes classifier [40], SVM [103] and these approaches report a detection rate of 70-80%.

Interactions with the System - Log Data

The user's interaction with the system, such as response time, response to the questions, questions difficulty level, number of hints used, number of wrong answers in a session, and goals achieved are stored in the log file. These data from the log the files are analyzed to detect the affective states. The features (data from the log files) are chosen based on the correlation, with the affective state measured using an independent method, or theoretical definitions of the affective states under consideration. The selected features are then applied to the classification algorithms to classify the affective states. The classifiers used in this approach are linear regressions classifier, Naive Bayes, decision tree, support vector machine and rule-based classifier. The research work in [61], models the user frustration based on the data from the search engine and reports the accuracy of 87% [61].

Multimodel Approach

In the multimodel approach, the researchers implement in a system, two or more of the approaches mentioned above and analyze the data to detect affective states. The general approach is to combine the data from the approaches mentioned above with the data from the physiological senors and create a model to detect affective states. There is a lot of research interest in combining data from different channels like facial expressions, body gestures and the log file to detect the users' affective states [89], [12]. The data from the posture sensor, log data, facial expressions and head gestures are combined and used to detect the affective states in [89], [2]. These studies report a higher detection accuracy compared to the detection of affective states using individual channels [89], [121].

Discussion

A comparative summary of the various approaches discussed so far to detect affective states is shown in Table 4.1. The comparison is based on factors such as degree of intrusion, cost, detection accuracy, and scalability. These factors are considered based on application of our research work to large classrooms. The factor "degree of intrusion" considers whether any external sensor is attached to the user's body or not. The factor "cost" considers the additional cost required to detect the users' affective states: Low - no extra cost, Medium - Less and one-time cost, High - relatively high cost and recurring. The factor "scalability" considers the feasibility of implementing the approach to a large number of students.

Table 4.1: Synthesis of Different Approaches to Detect Affect States

Data Used to Detect Affective States	**Intrusive**	**Cost**	**Detection Accuracy**	**Scalability**
Facial Expressions	No	Medium	Medium (70–80%) [58], [14], & [153]	High
Body Postures	No	Medium	Medium (70–80%) [116], [49]	Medium
Physiological Sensors	Yes	High	High ($>$ 80%) [159], [105] & [17]	Low
Voice and Text Input	No	Low	Medium (70–80%) [4], [40] & [103]	High
User Interactions	No	Low	High ($>$ 80%) [61]	High

In the approaches mentioned above, detecting affective state using physiological sensors is the best approach based on detection accuracy. However, implementing these approaches requires very high cost due to the sensors used; most of sensors require a large space to record the body signals from the user. Hence, most of the research studies in this approach are done in laboratory settings. Later, by invention of hand-held sensors like the skin conductance sensor, and pressure sensitive mouse, these approaches can be implemented in a real-world scenario. However, these sensors are intrusive to the users while they interact with the system. Moreover, the scalability of this approach to large class is very low due the cost and maintenance required. The facial expressions are tested using the facial expression videos from adults (> 18 years) [14], [153] and the user must face the video camera while working with the system. Also, this approach results comparatively less detection accuracy and has not been tested in the real-world environment. Voice and text data analysis is only applicable to the ITS which considers voice input or long text answers from the users. However, the Math ITSs, which is implemented and tested in schools (Active Math [112], Cognitive tutor [131], [6], Mathematics Tutor [151], [16], and PAT Algebra Tutor [130], [94]), neither uses voice nor long text as a input. Also, the research on detecting the affective state using voice and text data, reported a medium detection accuracy of 70 to 80 %. In these approaches, detecting the affective states by analyzing the user interactions is feasible in a large-scale, real-world scenario [133] and is non-intrusive to the users. In summary:

- Identifying affective states using the sensor signals is possible in laboratory settings, but difficult to implement at a large scale. Also, the physiological sensors are intrusive to the users.

- Facial analysis methods use a web-cam to analyze the facial expressions of the users. In the real-world scenario, keeping the camera in the right position, and expecting users to face the camera all the time is not feasible.

- Voice and text analysis methods can only be used in the ITS that considers voice and subjective answers as an input from the users.

In affective states, detecting and responding to the negative affective states are important as it might lead the student to drop out of learning [88]. Moreover, frustration is a cause of student disengagement and eventually leads to attrition [88]. Hence, in our research we focus on approaches that use student's interaction with the system to detect the students' frustration.

4.2 Detecting Frustration from the Student's Interactions with the System

In this section, we review the existing approaches related to our research on modeling and detecting frustration from the ITS log data. We review the systems which identify the students' affective states from the log files of the students' interactions as well as systems which do so by modeling the affective states. First we describe three systems–AutoTutor, a programming lab, and Crystal Island–which detect frustration based on data from the log file. Some of these systems also use biometric data, but we will focus only on detection approaches using log data. Later we discuss two systems–Crystal Island, and Prime Climb–which detects frustration by modeling the students goal and learning pattern.

AutoTutor [48] is a dialogue-based tutoring system. In AutoTutor, the students' affective states are detected based on the features from the log data. The affective states detected are frustration, boredom, flow and confusion. The features are identified based on the features from the log file, such as, response time, number of characters in a student's response, number of words in student's response, change in student's score and tutor feedback to a student's response. The data from 28 college students' interaction with AutoTutor were considered for the analysis. Large set of features were first identified from log data and then reduced from 17 to 11 by applying correlation analysis with affective states identified, using an independent method; the features which are significantly correlated were selected. To avoid redundancy among the selected features, dimensionality reduction techniques such as principle component analysis and linear discriminant analysis were used. The selected features were applied to a data-mining software Weka [108], to detect the affective states. The classifiers considered to detect affective states from Weka are:

- **Bayesian classifiers:** Naive Bayes and Naive Bayes Updatable.

- **Functions:** Logistic Regression, Multilayer Perceptron, and Support Vector Machines.

- **Instance based techniques:** Nearest Neighbor, K*, Locally Weighted Learning.
- **Meta classification schemes:** AdaBoost, Bagging Predictors, Additive Logistic Regression.
- **Trees:** C4.5 Decision Trees, Logistic Model Trees, REP Tree.
- **Rules:** Decision Tables, Nearest Neighbor Generalization, PART.

Independent methods used in this research to detect affective states and validate the results were self reporting, peer reporting and human observation of students' facial expressions. The reliability of human observers was measured in mutual agreement of affective state and the kappa value for agreement is K=0.36. Affective states were detected for every 20-second interval and the results are compared with the affective states from independent method. This study reports individual detection accuracy of boredom, confusion, flow, and frustration, when compared with neutral affect are 69%, 68%, 71%, and 78%, respectively. The AdaBoost classifier provided the best detection accuracy for boredom and confusion when compared to neutral. Simple logistic regression and C4.5 decision trees provided the best detection accuracy for flow and frustration respectively when compared to neutral affect. The features response time, tutor feedback, and change in student's score were correlated with the affective state frustration.

Crystal Island [111] is a task-oriented learning environment test-bed, featuring a science mystery in the domain of genetics. In Crystal Island, the students' affective state were predicted early based on the features from the log file and the data from physiological sensors. The affective states predicted in this system was frustration. The features considered to predict frustration were temporal features (such as time spent in a current location, and time spent after), locational features (such as current location, and past location), intentional features (such as goals being attempted, goals achieved and rate of goal achieved), and physiological response (such as blood volume and heart rate) from biofeedback apparatus attached to student hand. The data from 36 students' interaction with Crystal Island and biofeedback apparatus were considered for the analysis. This systems used two n-gram affect recognition models to early predict the frustration, a unigram model and bigram model. The features selected using n-gram models were applied to classifiers such as Naive Bayes, decision tree, and support vector machine to model frustration. The data-mining software, Weka [108] was used to predict frustration. Independent method used in this research to detect affective state was student self reporting. The student's were asked to report their affective states periodically (75 seconds) in the "self-report emotion dialog" box. The students were asked to select the emotion from set of six emotions; excitement, fear, frustration, happiness, relaxation and sadness. The best reported performance of this system [111] is an accuracy of 88.8%, precision of 88.7% and recall of 88.9% for predicting frustration using decision tree.

In a programming lab, in [132], the students' affective state in computer programming exercises across different labs is detected. The affective state detected in this system was frustration. The features considered to detect frustration were information from compiler data such as average time between compilations, total number of compilations, total number of errors and Error Quotient (EQ) construct. The error quotient construct is developed based on how students cope with syntax errors [86], this includes the number of pairs of compilations in error, number of pairs of compilations with the same error, number of pairs of consecutive compilations with the same edit location, and number of pairs of consecutive compilations with the same edit location. The data from ten students across five lab sections were collected for the analysis. The data mining tool Weka [108] was used to generate linear regression model to detect student average frustration. Independent method used in this research to detect affective states and validate the results was human observation of students' facial expressions. The reliability of human observers was measured using Cohen's kappa (k=0.65). Affective states were detected for every 20 second interval and the results were compared with the affective states from independent method. The study reports poor correlation coefficient r (r in the range of -0.3472 to 0.0136). The correlation coefficient, r reported across all five labs is r = 0.3178 [132]. The features which correlated with frustration are consecutive pairs of compilations with the same edit location, consecutive pairs of compilations with the same error, average time between compilations, and total errors.

In the above three systems, the feature set is constructed from the student interaction with the system (log data). In contrast, the following systems detect frustration by modeling the student's goal and the learning pattern, based on the theoretical definitions of frustration.

Affective state modeling in Crystal Island [139], creates a Dynamic Bayesian Network (DBN) model to capture the students' affective states. The affective states detected in this system were anxious, bored, confused, curious, excited, focused, and frustrated. The features considered to model students affective states are personal attributes of students which are identified from students' scores and personal surveys prior to interaction with system, observable environment variables such as goals completes, books viewed and successful tests, and appraisal variables which are student's cognitive appraisals such as learning focus and performance focus. The data from 260 students' interactions with Crystal Island was considered to model the affective states. Naive Bayes, Bayes net and dynamic Bayesian net were used to model the affective states. The independent method used in this research to validate the results is self reporting. The student's were asked to self-report their current mood in the dialog box for every seven minutes. The students were asked to select the emotion from set of seven emotions which are anxious, bored, confused, curious, excited, focused, and frustrated. The reported detection accuracy of all affective states is 72.6% using DBN. The individual detection accuracy reported for anxious, bored, confused, curious, excited, focused, and frustrated are 2%, 18%, 32%, 38%, 19%, 52% and 28%, respectively. The DBN for individual affect states were not discussed in this research study.

The Prime Climb is an educational game designed to help 6^{th} and 7^{th} grade students to practice number factorization. In Prime Climb students' emotions as described in OCC theory [120] were modeled using Dynamic Decision Network (DDN). The emotions from OCC theory considered in this system were joy, distress, admiration and reproach. The features considered to model the emotions were student goals (Have Fun, Avoid Falling, Beat Partner, Learn Math, and Succeed By Myself) captured using post-game questionnaire, personality types of students such as agreeableness which were identified using personality test, and interaction patterns such as move quickly, fall often, and ask advice often which are captured from log data. Sixty-six 6^{th} and 7^{th} grade students interaction with Prime Climb and data collected from survey questionnaire were considered to model the emotions. Dynamic Decision Network was used to model the emotions. The independent method used in this research to validate the results is self reporting. The students were asked to self-report their feeling about game and feeling about the agent. The students were asked to select a feeling from set of five options which are Very Bad, Bad, Neutral, Good, and Very Good. Self reporting dialog box was present permanently on the side of the Prime Climb game window. To collect more data, the self-report dialog box was designed to pop-up if the student was reporting his/her feeling for long time or the DBN model has detected a change in emotions. The emotions towards the self are joy and distress and the emotions towards the agent are admiration and reproach. The DBN detected the emotions based on whether the goals of the students were satisfied or not. The system [33] reports that the accuracy in detection of joy is 69%, distress is 70%, reproach is 47%, and admiration is 66%. The student feels joy when he achieves the goals and distress when he was not able to achieve the goal.

4.2.1 Discussion

Table 4.2 summarizes the research done in identifying frustration using the log data. The other affective states such as boredom, confusion were not detected in all the system described in this section systems hence we are reporting not them in this discussion. However, we note that the detection accuracy of other affective states are relatively equal to detection accuracy of frustration.

In the research studies mentioned above, AutoTutor, Crystal Island and Program-

Table 4.2: Research Works, that Identify Frustration Using the Data from Student Log File, with Number of Features, Detection Accuracy and Classifiers used

Ref Number	**ITS/Game used**	**Features used**	**Method of selecting the feature**	**Detection Accuracy**	**Classifiers used**
[48]	AutoTutor	Data from students' interaction	Correlation analysis	78%	17 classifier like NB, DT from Weka[108]
[111]	Crystal Island	Data from students' interaction and Physiological senors	All features	88.8%	NB, SVM, DT
[132]	Introductory Programming Course Lab	Data from students' interaction	Correlation analysis	Regression coefficient r=0.3168	Linear regression model
[139]	Crystal Island	Students' learning pattern and data from questionnaires	All features	28%	DBN
[33]	Prime Climb	Students' learning pattern and data from questionnaires	All features	For joy = 69% and for distress = 70%$^{\$}$	DDN

NB- Nave Bayes, SVM- Support Vector Machine, DT - Decision Tree, DBN - Dynamic Bayesian Network, DDN - Dynamic Decision Network, $ = this system was not detecting frustration

ming Lab detect the students' frustration based on the students' interaction with the system (log file). In these systems, frustration is detected using a combination of correlation analysis and theory. These three systems used a machine learning software called Weka [108], to create the model to detect affective states. The detection accuracy using the log data in these systems (AutoTutor [48] and Crystal Island [111]) is in the range of 77% to 88%. In the approach which models the affective states detects frustration and emotions based on the students' interaction pattern with the system and their goals. The reported detection accuracy for frustration [139] = 28% and for emotions [33] in the range of 47% to 70% which is comparatively lower than the systems which detects frustration using data from students interactions.

We note that in AutoTutor, Crystal Island and Programming Lab the frustration was detected in a fixed interval. These systems were not detecting frustration when the student's goal failed or other learning patterns. The interval where the affective state was detected, the student might be reading or doing calculations for his goal and not necessarily got response for his goal. However, Prime Climb system detects the students' emotion based on their goal and learning pattern and independent method to collect emotion is initiated for validation. In Prime Climb, theoretical definitions were used to model the affective states hence these systems capture not only the affective state but also the reason why the student is in that state. The reason identified by the system helps to respond to user's affective state based on the reasons for it. In our research, we focus on creating a model which has higher detection accuracy and also provides the reason why the student is frustrated.

4.3 Responding to Affective States

In this section, we focus on the systems (ITS and Educational games) which detect and respond to the students' affective states while they interact with the system. Responding to affective states in traditional teaching methods are not considered in this section. There are fewer research studies in responding to affective states than in detecting affective states in systems like educational games and ITS. We describe three systems in this section: affect-support computer games [93], Scooter the Tutor [133] and Wayang Outpost [164].

The affect-support computer game [93] induces frustration by freezing the screen when the students play the game. This system provides three types of response–ignore students' frustration, collect feedback from students, and provide messages. "Ignore students' frustration" response provides no motivational messages and does not collect feedback from the users. "Collect feedback" response collects the student feedback in terms of like how they feel and does not provide any motivational messages. In the third response, the system provides feedback messages and sympathy messages whenever the user reports frustration. The impact of these system responses provided to the students was analyzed using the data from 71 students. The system reports that in general students felt significantly less frustrated when playing the game without freezing than the game playing with freezing. The students who have got feedback message and empathy messages played the game significantly more time than the students' who got no messages. The study concludes that, any system can alleviate negative emotions by using empathy messages even if there is a source of negative emotions in the system.

Scooter the Tutor is an ITS to teach how to create and interpret scatter plots of data [133]. This system detects when the students game the system, using the student's interaction with the system. If the student games the system, the agent (scooter) will show empathy towards the student with motivational messages like "student should work in order to learn." If the student does not game the system, the scooter will show the happy expressions and occasionally provide positive messages. This system was tested using 124 students. The students were divided into control and experimental groups. The students' affective states like boredom, frustration and confusion were observed using human observers while they interacted with the ITS. This system reported no significant difference between the control and experimental groups in the observed affective states. However, when considering only the number of frustration instances, the control group felt more number of frustration instances than experimental group. In general, the students reported that they liked the agent.

Wayang Outpost [164], is an interactive math ITS used to teach math to school students. This system used physiological sensors to detect affective states. The data from the sensors were applied to data-mining tools to detect the affective states. The affective states considered in this research were confidence, excitement, boredom, focus, frustration, and anxiety. Based on the detected affective states, this system provides delayed or immediate feedback. The agent reflects the student's emotion when it detects the affective state and motivational messages were provided. The motivational messages were created based on Dweck's [54] motivational message theory–praise the effort instead of the result. The preliminary analysis of this system is tested using 34 students' interaction with

Wayang Outpost. This system reported that the students change their engagement (from low to high) with the system based the interventions such as tips to read the problem and hint carefully, performance graph of last fix questions and asking student to think about the problem.

4.3.1 Discussion

Table 4.3 summarizes the research done in responding to affective states.

Table 4.3: Related Research Works to Respond to Student's Affective States along with the Theories used, Experiment Method and Results

Ref Number	ITS/Game used	Theory used to respond to frustration	Experiment Method	Results
[93]	Affect-Support computer game	Active listening, emotional feedback, sympathy statement [118]	Factorial study, 2 (level of frustration) x 3 (interactive design), N = 71. Self reporting using questionnaire	On an average the affect support group played more minutes compared to non-affect support group.
[133]	Scooter the Tutor	Agents were given emotions	Control-experiments group study. N = 60. Human observation	Reduction in frustration instances. There is no significant difference in observed affect between control and experimental group.
[164]	Wayang Outpost	Agent to reflect student's affective states and messages based on Dweck's messages [53], [54]	N = 34, physiological sensor data to detect affective states	Initial studies results that students change their behavior based on digital interventions

N = Number of participants

The systems discussed in this section respond to the student's frustration by showing motivational messages and reflecting their emotions using agents. Three different approaches (self reporting, human observation and physiological sensors) are used to detect affective states in these systems. These systems detect only the affective state not its reasons; hence, they provide motivational messages which respond to frustration in general and may not address the specific cause which induces frustration. These systems report that responding to frustration positively influences the student's learning–in areas such as increased time of playing the game, fewer frustration instances or change in student engagement with the system. Our research focus is to respond to the reasons for frustration using the motivational messages.

4.4 Motivation for Our Work

In this chapter we described our literature survey on detecting and responding to the affective states of the users. First we described the existing approaches to detect affective states in educational systems like Games, Intelligent Tutoring Systems (ITS) and search engines. Based on our synthesis we focused our literature survey on detecting the affective states by analyzing the log data, since it is a feasible approach in a large-scale, real-world scenario and non-intrusive to the students. We then discussed the systems which detect frustration based on the users' interaction with the system (log file). The existing systems predict frustration at different intervals and the reasons for the frustration are not explored. The system which predicts frustration based on theoretical definitions performed poorly compared to the system which detects frustration using data from log file. Hence, their is a need for a frustration model with higher detection accuracy and also with the reason why the student is frustrated. Later, we discussed the systems which respond to the students' frustration by motivational messages and reflecting students' affective states using agents. The results reported in these systems conclude that the systems which respond to frustration by motivational messages positively influences the students' interaction with the system. However, these systems detect only the affective state not its reasons; hence, they provide motivational messages which respond to frustration in general and may not address the specific cause which induces frustration. Hence, we require a system which address the students' frustration.

In our research, we design the model to detect frustration based on the students' learning patterns and goals applied to the students' interactions (log file). We applied the theoretical definitions of frustration to the students' interactions (data from log file) to construct the features, by doing this we constructed the features which captures the students' goals and learning patterns. These features are used to create a frustration model, which detects student's frustration when the student receives the response about his/her goals. We used the human observation data from students' facial expressions as a independent method to train and validate our model. The reliability of human observers are established using Cohen's Kappa. Since, our frustration model is designed based on the students' goals and learning patterns, the reason why the student is frustrated is known. Later, we developed an approach to respond to reasons for frustration. Our approach is build on the strategies used in existing system to respond to frustration.

Chapter 5

Detect Frustration

Affective computing is concerned with detecting and responding to the affective states. Our research aim is to create a model to detect and respond to frustration accurately in real-time when students are working with an ITS. In this chapter we describe our approach to detect student's frustration when they interact with the ITS. Then we discuss, how our approach is applied to an ITS (Mindspark), and its performance compared to other data-mining approaches. The frustration model is created by constructing features from the ITS log data based on theoretical definitions of frustration. We focus on the instances of frustration that occur due to goal blockage. Frustration in this case is considered to be a negative emotion, as it interferes with a student's desire to attain a goal [115]. To model the frustration we considered different classifiers used in existing approaches, including dynamic Bayesian nets; however, we start with the linear regression classifier because it is easy to build and it is a simple classification model to start with. The linear regression model is flexible, fast and accurate when cross-validated [29]. Also the linear regression classifier model, informs us of the factors contributing to frustration. It helps us to determine which features contribute most to frustration, as well. Thus the linear regression model can help us to respond to frustration systematically, and identify potential sources of frustration, thereby, helping students to avoid it. However, later we apply our constructed features to other classifiers such as decision tree and support vector machine.

5.1 Theory-Driven Model to Detect Frustration

In this section, the theory-driven approach to detect frustration is described. The sequence of steps to model frustration is shown in Figure 5.1. The selection and combination of features from the ITS log file is done through a systematic process based on an analysis of goal-blocking events. Guided by the theoretical definition (Step 1), we first identify the goals of the student with respect to their interaction with the ITS, and select the top n goals (Step 2). Based on information from the student log, a blocking factor, bf, for each of the n goals is identified (Step 3). For example, $goalj.bf$ represents the blocking factor for the $goalj$. We formulate a linear model for F_i, the frustration index at the i^{th} question based on the blocking behaviors of student goals (Step 4). Since the features in linear regression model are constructed based on the theoretical definition of frustration, we call this approach as theory-driven approach. We apply a threshold to the frustration index F_i, to detect whether the student is frustrated or not. The average of values used to represent frustration and non-frustration, during the training process, is used as threshold. The analysis on different threshold values is explained later in this chapter. The weights of the linear regression are determined during the training process (Step 5)–with labeled data from human observation–which is an independent method to identify affective states. The performance of the model is validated by detecting frustration in the test data and comparing results with independent method (Step 6).

The proposed theory-driven linear regression model to detect frustration is given below:

$$\begin{aligned} F_i = \alpha[w_0 + w_1 * goal1.bf + w_2 * goal2.bf + ... \\ +w_n * goaln.bf + w_{n+1} * t_i] + (1-\alpha)[F_{i-1}] \end{aligned} \tag{5.1}$$

The weights $w_0, w_1, w_2, ..., w_n$ in the equation above are determined by linear regression analysis–explained later in this chapter. As explained in the previous paragraph, the terms $goal1.bf$, $goal2.bf$, . . . , $goaln.bf$, are the blocking factors for goals $goal1$,

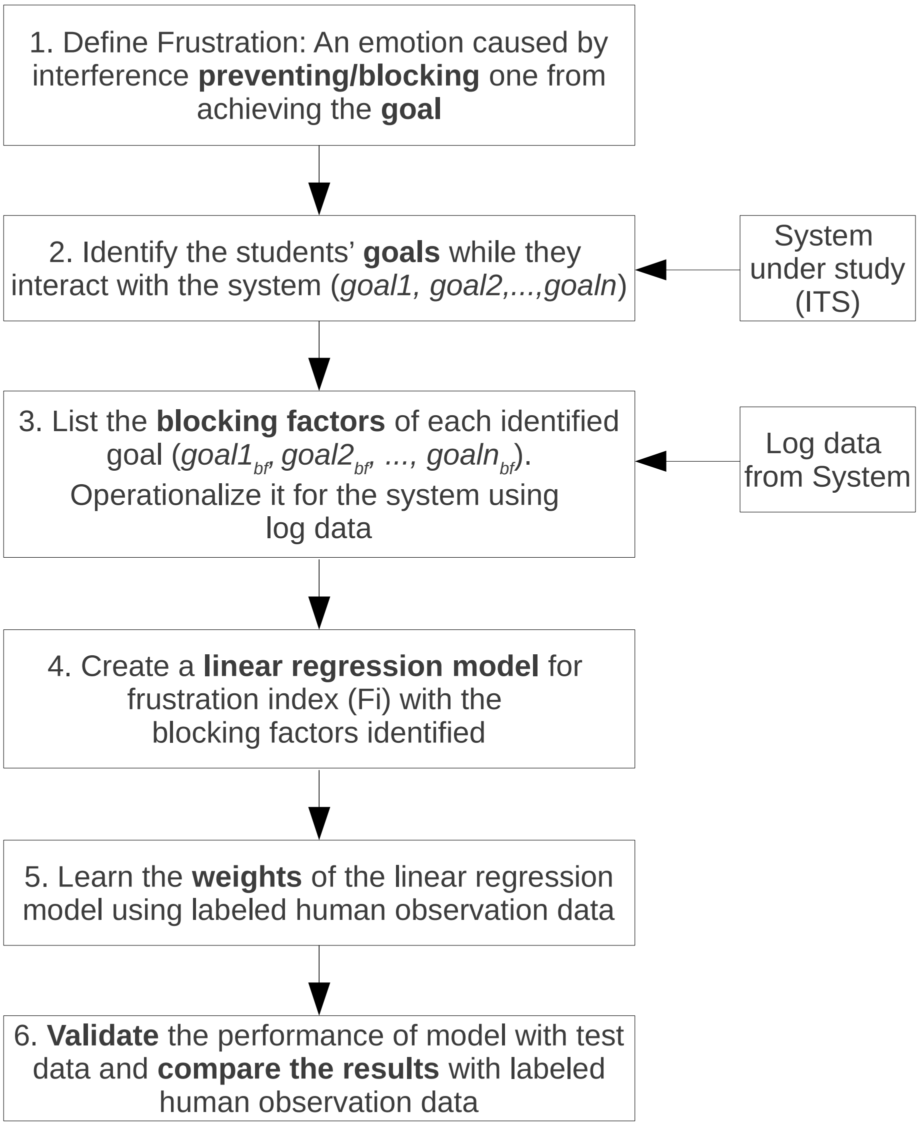

1. Define Frustration: An emotion caused by interference **preventing/blocking** one from achieving the **goal**
2. Identify the students' **goals** while they interact with the system (*goal1, goal2,...,goaln*)
System under study (ITS)
3. List the **blocking factors** of each identified goal (*goal1*$_{bf}$, *goal2*$_{bf}$, ..., *goaln*$_{bf}$). Operationalize it for the system using log data
Log data from System
4. Create a **linear regression model** for frustration index (Fi) with the blocking factors identified
5. Learn the **weights** of the linear regression model using labeled human observation data
6. **Validate** the performance of model with test data and **compare the results** with labeled human observation data

Figure 5.1: Steps of the theory-driven approach to create a frustration model using data from the log file

$goal2, \ldots, goaln$, respectively. The term t_i, the time spent by the student to answer the question i., is included on the basis of work done by Lazara et al. [98], which states that time spent to achieve the goal is an important factor in frustration. The last term in the equation, $(1-\alpha)[F_{i-1}]$ accounts for the cumulative effect of frustration. We include this term on the basis of [93], which states that frustration is cumulative in nature. The value of α, determines the contribution of frustration at $(i-1)^{th}$ question to frustration at i^{th} question, α which ranges from 0 to 1. We assume that the student is not frustrated at the beginning of their interaction with the ITS, and hence, choose $F_i = 0$ for $i = 1, 2, 3$.

We restrict the scope of our approach to identify frustration that occurs due to students' goal blockage while interacting with the ITS. We do not include frustration that might have occurred due to other factors external to students' interactions with the ITS. Hence, we are primarily concerned with how correct our detection (precision) is, even though we might not be able to find all the frustration instances that exist (recall).

5.2 Theory-Driven Model for Mindspark Log Data

In this section we explain the application of the theory-driven approach to the Mindspark log data. Our research goal is to detect frustration of the students while they interact with the Mindspark.

5.2.1 Frustration Model for Mindspark

We create the linear regression model based on steps given in Figure 5.1.

Step 1. Frustration definition: We begin with the definition of frustration from theory as "an emotion caused by interference preventing/blocking one from achieving the goal". The definition of frustration was discussed in Chapter 2 and it is reproduced below:

- Frustration is the blocking of a behavior directed towards a goal [115].
- The distance to the goal is a factor that influences frustration [97], [149].
- Frustration is cumulative in nature [137], [50].
- Time spent to achieve the goal is a factor that influences frustration [98].
- Frustration is considered as a negative emotion, because it interferes with a student's desire to attain a goal [124].

Step 2. Students' Goals: We identified the four most common goals of students while interacting with Mindspark. To identify these goals, we conducted interviews with the staff of EI-India. The staff of EI-India conduct student interviews every month, as part of their job. We requested them to interview the students, about their goals while they interacted with the Mindspark. The interview was informal and conversational. The questions emerged from the immediate context and were based on the student's answers; there were no predetermined questions. Eight students were interviewed by random selection, during their interaction with Mindspark. The interview details are given below.

Student Interview Details

- Goal: To identify the students' goals while interacting with the Mindspark.
- Class: Sixth standard in Indian Certificate of Secondary Education (ICSE)* board.
- Subject: Student's were learning Maths using Mindspark.
- Location: Mumbai, India.
- Time per student: Three to five minutes, and five questions per student.
- Data Collection: Voice recorder was used to record the interaction with the students.

We transcribed the interviews, and analyzed the transcripts to identify the goals of the students. The important goals of the student were collecting Sparkies (reward points in Mindspark), completing the current question correctly and quickly completing the topic. These goals and corresponding blocking factors are discussed in next step.

Step 3. Blocking Factors: The goals $goal1$, $goal2$, $goal3$, $goal4$ and the corresponding blocking factors $goal1.bf$, $goal2.bf$, $goal3.bf$, $goal4.bf$ are given in the Table 5.1. To model the blocking factor (bf) of each goal, we consider students' response to Mindspark questions, a feature captured in the Mindspark student log file.

For $goal1$ of "*to get the correct answer to the current question*" the blocking factor is getting the wrong answer to the current question. We use a_i to represent the answer

*http://www.cisce.org/

Table 5.1: Student Goals and Blocking Factors for Mindspark

Student Goal	**Blocking factor**
$goal1$: To get the correct answer to the current question	$goal1.bf$: Answer to the current question is wrong
$goal2$: To get a Sparkie (answer three consecutive questions correctly)	$goal2a.bf$: Answers to two previous questions are correct and to the current question is wrong $goal2b.bf$: Answer to the previous question is correct and to the current question is wrong
$goal3$: To reach the Challenge Question (answer five consecutive question correctly)	$goal3a.bf$: Answers to four previous questions are correct and to the current question is wrong $goal3b.bf$: Answers to three previous questions are correct and to the current question is wrong
$goal4$: To get the correct answer to the Challenge Question	$goal4.bf$: Answer to the Challenge Question is wrong

to the current question; $a_i = 1$ if correct, $a_i = 0$ if wrong. The blocking factor of $goal1$ is captured using:

$$goal1.bf = (1 - a_i) \tag{5.2}$$

For $goal2$, "*to get a Sparkie*" the student should answer three consecutive questions correctly. This goal can be blocked, if a student gets any question wrong in a sequence of three questions. Since the blocking factor by getting the wrong answer to the current question is already addressed in $goal1.bf$, we consider only the blocking factor by getting the second and third answer wrong, in a sequence of three questions. Hence $goal2.bf$ has two components. One way in which $goal2$ can get blocked is, if the student answers first two questions correctly in a sequence of three questions and the third question wrongly. This is captured by blocking factor $goal2a.bf$:

$$goal2a.bf = (a_{i-2} * a_{i-1} * (1 - a_i)) \tag{5.3a}$$

The second way in which $goal2$ can get blocked is, if the student answers only the first question correctly in a sequence of three and the second question wrongly. This is captured by blocking factor $goal2b.bf$:

$$goal2b.bf = a_{i-1} * (1 - a_i) \tag{5.3b}$$

The blocking factor of $goal2$ is:

$$goal2.bf = goal2a.bf + goal2b.bf \tag{5.4}$$

For $goal3$, "*to reach the Challenge Question*", the student should answer five consecutive questions, correctly. This goal can be blocked, if the student gets any question wrong in a sequence of five questions. Since the blocking factor by getting the wrong answer to the first, second and third question in a sequence of five questions is already addressed in $goal1.bf$, and $goal2.bf$, we consider only the blocking factor by getting the fourth and fifth answers wrong in a sequence of five questions. Hence, $goal3.bf$ has two components. One way in which $goal3$ can get blocked is, if the student answers only first four questions correctly, in a sequence of five questions and the fifth question wrongly. This is captured by $goal3a.bf$:

$$goal3a.bf = (a_{i-4} * a_{i-3} * a_{i-2} * a_{i-1} * (1 - a_i)) \tag{5.5a}$$

The second way in which $goal3$ can get blocked is, if the student answers only first three questions correctly, in a sequence of five questions and the fourth question wrongly. This is captured by $goal3b.bf$:

$$goal3b.bf = (a_{i-3} * a_{i-2} * a_{i-1} * (1 - a_i)) \tag{5.5b}$$

The blocking factor of $goal3$ is:

$$goal3.bf = goal3a.bf + goal3b.bf \tag{5.6}$$

For $goal4$: "*To get the correct answer to the Challenge Question*" the blocking factor is getting the answer to Challenge Question wrong. The blocking factor of $goal4$ is captured using:

$$goal4.bf = I * (1 - a_i) \tag{5.7}$$

where, I is the indicator for whether the current question is the Challenge Question or not. $I = 0$ for normal question and $I = 1$ for Challenge Question.

Step 4. Linear Regression Model: The mathematical model to detect frustration for the Mindspark data is given in Equation 5.8, with the individual terms $goal1.bf$, $goal2.bf$,...,$goal4.bf$, being defined in Equations 5.2-5.7:

$$\begin{aligned} F_i = \alpha[w_0 + w_1 * goal1.bf + w_2 * goal2.bf + w_3 * \\ goal3.bf + w_4 * goal4.bf + w_5 * t_i] + (1 - \alpha)[F_{i-1}] \end{aligned} \tag{5.8}$$

Step 5 and Step 6 of Fig. 5.1 are explained later in this Chapter.

5.2.2 Feature Selection and Combination in the Theory-Driven Approach

The features (that is, the goal-blocking factors) in our linear regression model for the frustration index F_i, are not the features directly available in the log file. Instead, they are created by selecting and appropriately combining the available features from the Mindspark log file (Table 5.2) using theory. We illustrate the advantages of selecting features by applying a goal-blockage based theory, by comparing a feature with a simple combination of data from the log file. Simple combination would be the most obvious way and the first step in creating features from the log data. Both methods start with the same raw data–features from the log file which indicate students' response to the current question (a_i), and the last two questions (a_{i-1}) and (a_{i-2}).

In our approach, we use a complex combination of the previous three responses ($goal2.bf$ in this example). This combination, $goal2_bf$, is calculated using equation (5.4). If $a_i = 1$, indicates that the current question is correct and $a_i = 0$ indicates that the current question is wrong. When the goal of achieving a Sparkie ($g$oal2), is blocked near the goal (student answers first two questions correctly, in a sequence of three questions), then the blocking factor $goal2.bf$, which contributes to the frustration index is high. Similarly, when the goal is blocked midway to the goal (student answers only the first of a sequence of three questions correctly) then the value of $goal2.bf$ is medium. Finally, when the goal is blocked far from the goal (student answers the first of a sequence of three questions wrongly), the $goal2.bf$ has a low value. The difference between goal-blocking near the goal and far from the goal is captured in our approach which is not possible in simple combination of the data from the log file (the last column in Table 5.2).

Table 5.2: An example to illustrate the advantages between selecting features by applying goal-blockage based theory and by a simple combination of data from the log file when applied to Mindspark

			Theory-Driven Approach			**Simple Combination of Data from Log File**
a_{i-2}	a_{i-1}	a_i	eq 5.3a	eq 5.3b	eq 5.4 ($goall2.bf$)	Sum of last 3 responses
0	0	0	0	0	0	0
0	0	1	0	0	0	1
0	1	0	0	1	1	1
0	1	1	0	0	0	2
1	0	0	0	0	0	1
1	0	1	0	0	0	2
1	1	0	1	1	2	2
1	1	1	0	0	0	3

In the Table 5.2, the goal2.bf is to capture whether the goal of getting a Sparkie is blocked or not. As explained in the previous paragraph the goal can be blocked by two factors. For the response pattern (0,0,0) the student's goal to get a Sparkie is neither started nor blocked hence the blocking factor for goal2 ($goal2.bf$) is 0. In the second response patter (0,0,1) the student's goal to get a Sparkie is started at the current response but not blocked. Hence goal2.bf = 0. In third response pattern (0,1,0) the student's goal of getting a Sparkie is started in the last question but blocked due to incorrect answer in the current question. Hence the $goal2.bf$ is 1 (midway to the goal). In the fourth response pattern (0,1,1) the goal of getting a sparking is started in the last questions and not blocked, hence there is no blocking factor to this goal, $goal2, bf = 0$. Similarly the value of $goal2.bf$ for other response patterns are calculated based on the blocking factors of the goal not by a combination of last three questions. Since the reasons for goal blocking are identified in the finer detail, it may help us in avoid the negative consequences of frustration in its detailed aspects–facilitating the prevention of frustration instances.

5.3 Experiment Methodology

In this section, we discuss the independent method we used to identify frustration: human observation. We conducted human observations of students interacting with the Mindspark to validate our frustration model and to determine the weights of our model. We describe the observation details, labeling of the observations and the metrics used to compare the results of our frustration model with those from the human observation method.

5.3.1 Human Observation

The goal of human observation was to observe the students' facial expressions and label them as frustrated or non-frustrated. In the human observation method, the observer observes a student's facial expression while s/he interacts with the ITS. In this subsection, we discuss the expressions observed and affective states identified from facial expressions.

5.3.1.1 Sample

We recorded videos of 27 students' (13 female, 14 male) facial expressions; six students from a school in Mumbai, India and 21 students from a school in Ahmedabad, India (both, urban areas in India). The students were fifth or sixth standard students in the age range of 10-12. We recorded the video of students' expressions while they interacted with a Mindspark during a session in their school. Each video was 25-30 minutes, and on average, it contained a student's facial expressions over 30 to 40 questions.

5.3.1.2 Video Recording Procedure

We contacted two schools which were already using Mindspark in their curriculum. We explained our objectives to students and their parents, and requested their consent to record students' facial expressions. The recording of the facial expression and the storing of the videos adhered to ethics committee guidelines. The students' facial expressions were recorded using a webcam and the students' interaction with the computer were recorded using Camstudio[†]. The collected videos were used for the human observation. The students' facial expressions were coded, after they received feedback to the response they submitted for each question. Our goal was to capture students' expressions at the point where they learn whether their answer to Mindspark's question was correct or wrong.

5.3.1.3 Instrument

We used an observation protocol based on a facial analysis coding system [87], [152]. The data collection sheet contained the following information: Student ID (Mindspark login ID), Question Number and observation made by the observer. A sample observation that records a student's expressions is given in Table 5.3.

[†] www.camstudio.org

5.3.1.4 Observation Procedure

The observers were Ph.D. students in Educational Technology at the Indian Institute of Technology (IIT) Bombay. All observers had taken the 'Research Methodology in Education' course, and hence had an understanding of the observation data collection method. All the observers had practiced facial expression coding [87], [152] during a pilot observation, prior to the actual study. After the pilot study, the observations were checked for inter-observer reliability. The observers agreed eighty percent of the time with other observers' facial expression coding and Cohen's κ was found to be 0.74–a substantial agreement in the pilot study. In our actual study, the observers observed students' facial expression from the video whenever the student submitted an answer to a question. The recorded videos helped the observers to pause the video and note down the expression. The observations were recorded on the data collection sheet. In our study, only one observer out of four was involved in building the model. Moreover, the student's interaction with the computer is used only to identify when to observe the facial expressions and is not shown to the observer during observation process. Hence, the observation process is truly an independent method of identifying frustration.

The facial expression observed are:

Face: Inner eyebrow rise, outer eyebrow rise, brow raise, brow lower, eyes wide open, eyes closed, mouth open, jaw drop, mouth funnel-rounding, tongue show, teething, smile, lip tightener and blink.

Head Movement: Left (with reference to the student), right, head up, head low, head forward, head backward, head shake up and down and head shake side to side.

Speech: all words, for example *yes*, *no*, *shhh* and other were noted.

Hands Gesture: Hand on chin, scratching of head, tightening of fingers and raising of fist.

Others: Leaning forward, relaxed, sitting at edge of the seat, mouse tapping, keyboard banging and chair movement.

5.3.1.5 Labeling Technique

The observers classified the students' facial expressions into frustration and non-frustration, based on the guidelines given in [46], [134] and [71].

The key behaviors to express frustration are:

- Outer brow raise
- Inner brow raise
- Pulling of hair
- Statements like "what", "this is annoying", "arey (Hindi word for expressing disappointment)"
- Banging on the keyboard or the mouse
- Cursing

We show labeling of the students' affective states along with sample observation data in Table 5.3.

5.3.1.6 Universality in Facial Expressions

Facial emotions have been reported by Ekman to be universal [56]. The study by Hillary et al., [7] reviewed 87 articles, related to cross-culture facial expression, and reports that

- Certain core components of emotions are universal
- "Emotions may be more accurately understood when they are judged by members of same national, ethnic or regional group that had expressed the emotion"

The study by MK Mandal et al. checks the observer-expressor culture differences. The study in [107] includes 43 university students from Canada and 43 university students from India. All the participants were asked to recognize the emotions from facial expressions. The study reports that:

Table 5.3: Sample Human Observation Sheet to Record Students' Facial Observations and to Label it as Frustration (Frus) or Non-Frustration (Non-Frus)

Question Number	Observation	Classification
1	Mouth open, lower lip down	Non-Frus
2	Reading aloud	Non-Frus
3	Making noise (cha, arey), resting forehead with fingers	Frus
4	Lips tightening	Non-Frus
5	Two hands raised up to to chest	Non-Frus
6	Hands clamped	Non-Frus
7	Reading	Non-Frus
8	Head lean towards screen immediately	Non-Frus
Challenge Question	Inner brow up, eyes wide open, noise (hindi cursing words)	Frus
9	No expression	Non-Frus
10	Mouth little open, eyes wide open	Non-Frus
11	Inner eye brows lowered (shrink)	Non-Frus

- "Happy, sad, and disgusted expressions are judged in the same way regardless of the culture of the expressor or that of the observer"

- For surprise and anger there is a mismatch if the observer-expressor culture differs. However, these emotions are correctly classified if the observer-expressor are the from same culture. This supports the argument by Hillary et al., [7].

Based on the research results reported in the above articles we conclude that, (i) the facial expressions are universal for frustration [56], and (ii) emotion recognition from facial expression is done better if the observer-expressor are from the same culture/nation (which is the case in our study). Hence, our facial expression observation is a valid independent analysis to validate our model.

5.3.2 Analysis Procedure

We recorded 932 observations from 27 students. Among those, 137 observations were classified as frustration (Frus) and remaining as non-frustration (Non-Frus). The dataset is stratified at the student level. We represent the values obtained from human observation as B_i at the i^{th} instance, $B_i = 0$ for non-frustration and $B_i = 1$ for frustration.

To validate our model, we need to calculate the Frustration Index, F_i (using Equation 5.8), for the students whose facial expressions were observed. In order to calculate F_i, the weights of the frustration model need to be learned. The procedure to learn the weights of our frustration model, corresponding to **Step 5** in Fig. 5.1, is given below:

1. To maintain a uniform scale among all the features in the frustration model, we apply normalization to all the features. We used the following normalization equation.

   $$X_{new} = \frac{X - Mean(X)}{Max(X) - Min(X)}$$

   Here, X_{new} is the normalized value of feature X. We used the range ($Max(X) - Min(X)$) of the feature in the denominator, instead of standard deviation. This is because the feature "response time" had a wide range of data (from 1 second to 200 seconds) compared to other features. Hence standard deviation in the denominator will not normalize all the features in a uniform scale.

2. The data from the Mindspark log file is divided into two groups–training and testing–to create and validate our frustration model, respectively. We use cross-validation technique [66] to check how generalizable our frustration model is, when we apply it to independent data. In this report, we used a tenfold cross-validation technique [66] for all the experiments. In tenfold cross-validation the data is divided into 10 subsets. The model is trained using the nine subsets of data and tested on the remaining data. This process is repeated to test all sets of data. We divided the data based on number of students.

3. We use linear regression analysis to identify the values for weight by assigning, 0 and 1 to represent non-Frus and Frus, respectively, in the training dataset.

4. We apply threshold to F_i, to classify the frustration index values as frustration and non-frustration; we call this value as predicted value P_i.

5. We use the trained model (weights learned from linear regression analysis), to detect frustration on the test dataset. Compare our detection with human observation.

5.3.3 Metrics to Validate Frustration Model

The metrics used to measure the performance and to validate our frustration model are discussed in this subsection. Based on the feature from the log file, we detect students' affective state at a given instant, as frustration or non-frustration; hence, we consider it as a binary classification problem. In binary classification, most of the evaluation metrics are based on a contingency table [140], [165]. The 2x2 contingency table for our research is shown in Table 5.4:

Table 5.4: Contingency Table

	Actual Frus	Actual Non-Frus
Pred Frus	True Positive (TP)	False Positive (FP)
Pred Non-Frus	False Negative (FN)	True Negative (TN)

where, Actual Frus and Actual Non-Frus are results from human observation based on students' facial expressions while they interact with the ITS. Pred Frus and Pred Non-Frus are detected values of frustration and non-frustration, respectively, using our frustration model on the data from the Mindspark log file.

- *True Positive (TP)* is the number of frustration instances correctly detected as frustration.

- *True Negative (TN)* is the number of non-frustration instances correctly detected as non-frustration.

- *False Positive (FP)* is the number of non-frustration instances detected as frustration.

- *False Negative (FN)* is the number frustration instances detected as non-frustration.

The most common metric used in the classification problem based on contingency table is accuracy [113].

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{5.9}$$

Accuracy measures the ratio of correctly detected instances (TP + TN) to total instances.

Other standard measures in classification problems are precision and recall based on the contingency table. Precision measures the ratio of correctly detected frustration instances (TP) to the total number of detected frustration instances (TP + FP).

$$Precision = \frac{TP}{TP + FP} \tag{5.10}$$

Recall measures the ratio of correctly detected frustration instances (TP) to the actual number of frustration instances identified from human observation (TP + FN).

$$Recall = \frac{TP}{TP + FN} \tag{5.11}$$

Since our dataset is an unbalanced distribution of frustration and non-frustration, we calculate the F1 score and Cohen's kappa to measure the performance of our model compared to random guesses.

As mentioned earlier, it is not feasible to detect all types of frustration experienced by a student from all sources, especially those extrinsic in nature. In our research, we are interested in detecting students' frustration arising from their interactions with the ITS. Hence, we report precision and recall of detecting frustration (instead of detecting non-frustration). Our goal is to ensure the correctness of our detection of frustration, instead of being able to detect all frustration instances encountered by students while interacting with the ITS. Hence, high precision, and not high recall, is the important metric.

5.4 Results

In this section we describe the performance of our frustration model. In the frustration model, our goal is to minimize the difference in detected frustration P_i and corresponding human observation values B_i, $(P_i - B_i)$

$$min(P_i - B_i)^2$$

$$\text{by varying } w_0, w_1, w_2, w_3, w_4, w_5$$

To solve the above problem, we used a GNU Octave[‡], an open source numerical computation software. The values of weights W_0 to W_5 are assigned equally during the training phase of our model. This indicates that all the goals are assigned equal weights during training of our model. In our experiments, to train the weights to optimum value, we used the gradient descent algorithm with step size = 0.001. Our approach leads to a converged set of weights after 70000 iterations, as seen in Fig. 5.2. Although our data set is not large, to generalize our linear model we used gradient descent algorithm to find the global optimum value for our weights.

The frustration model with trained weight values is shown in Equation 5.12. The features are normalized as explained in the previous section.

$$\begin{aligned} F_i = 0.8[0.147 + \ 0.423 \ * \ Feature1 - \ 0.0301 \ * \ Feature2 \\ + 0.0115 \ * \ Feature3 + \ 0.8359 \ * \ Feature4 \\ + 0.1864 \ * \ Feature5] \ + \ 0.2 \ [F_{i-1}]. \end{aligned} \tag{5.12}$$

To choose the best alpha value we find the performance of our model for different

[‡] http://www.gnu.org/software/octave/

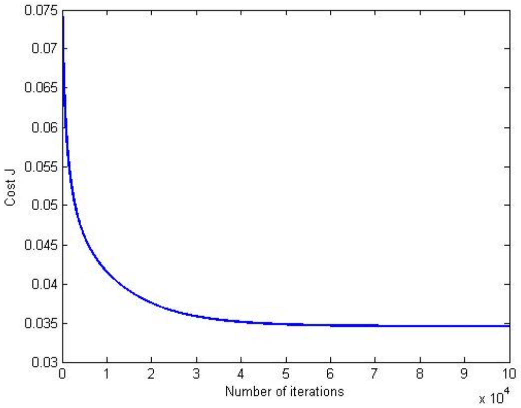


Figure 5.2: Convergence of Weights of our Linear Regression Frustration Model using Gradient Descent Algorithm. The Cost J in the Figure is the error function (difference between detected observation P_i and human observation B_i)

alpha values from 0.3 to 0.9. The results of frustration model for different alpha values are given in table 5.5.

Table 5.5: Frustration model performance for different alpha values

Alpha	TN	TP	FP	FN	Precision	Recall	Accuracy	Kappa
0.3	0	0	137	795	0.00%	0.00%	85.30%	0.0000
0.4	2	1	135	794	66.67%	1.46%	85.41%	0.0224
0.5	5	2	132	793	71.43%	3.65%	85.62%	0.0560
0.6	41	8	96	787	83.67%	29.93%	88.84%	0.3939
0.7	44	10	93	785	81.48%	32.12%	88.95%	0.4118
0.8	45	12	92	783	78.95%	32.85%	88.84%	0.4132
0.9	45	13	92	782	77.59%	32.85%	88.73%	0.4099

From the analysis it is clear that alpha = 0.8, gives better performance based on kappa values. Hence, we use alpha = 0.8. To determine the threshold value, which classifies frustration and non-frustration in our model, we calculated the average of the values we used to represent frustration and non-frustration in the training dataset. The threshold value used is 0.5 (average of 0 and 1). Since we used the linear regression classifier, we consider the mid-value as the threshold. However we find the performance of the model for different threshold values from 0.2 to 0.9. The results of the frustration model for different threshold values are given in table 5.6.

Table 5.6: Frustration model performance for different threshold values

Threshold	TN	TP	FP	FN	Precision	Recall	Accuracy	Kappa
0.2	134	141	3	654	48.73%	97.81%	84.55%	0.5652
0.3	130	132	7	663	49.62%	94.89%	85.09%	0.5683
0.4	79	64	58	731	55.24%	57.66%	86.91%	0.4873
0.5	45	12	92	783	78.94%	32.85%	88.84%	0.4132
0.6	38	5	99	790	88.37%	27.74%	88.84%	0.3786
0.7	20	5	117	790	80.00%	14.60%	86.91%	0.2111
0.8	2	0	135	795	10.00%	1.46%	85.52%	0.0247
0.9	2	0	135	795	10.00%	1.46%	85.52%	0.0247

We choose the threshold value based on the kappa value. From the threshold analysis table, we observe that the kappa values are best for threshold values 0.2, 0.3, 0.4, and 0.5, however the precision value is very low compared to the threshold value = 0.5. The threshold value of 0.5 gives the best precision and recall combination, based on our research interest. Our research interest is to achieve better precision than recall. Hence we choose the average of 0 and 1 (values assigned for frustration and non-frustration, respectively, to train the model) = 0.5 as threshold for our frustration model. The selected threshold gives a better balance of precision and recall compared to other threshold values.

The performance of our frustration model on Mindspark data, using tenfold cross-validation, compared to human observation is given in Table 5.7.

Table 5.7: Contingency Table of Our Approach when Applied to Mindspark Log Data

		Human Observation	
		Frustrated	**Non-Frustrated**
Pred	**Frustrated**	45	12
Result	**Non-Frustrated**	92	783

The values from Table 5.7 are used to calculate the performance of our model. The results are given in Table 5.8.

Table 5.8: Performance of our Approach Shown Using Various Metrics when Applied to Mindspark Log Data

Metrics	**Results**
Accuracy	88.84%
Precision	78.94%
Recall	32.85%
Cohen's kappa	0.41
F1 Score	0.46

From the above results, the accuracy and precision of our frustration model are high compared to recall. As we mentioned earlier, we are interested in how correct our detection of frustration is (measured by Precision) instead of detecting all frustration instances encountered by students (reflected by Recall). Thus, the results are aligned with our research goals. The Cohen's kappa (0.41) and F1 measure (0.46) are found and it is an acceptable value from affective state detection in ITS.

5.5 Performance of Data-Mining Approaches Applied to the Data from Mindspark Log File

While the theory-driven approach gave high accuracy and precision for detecting frustration, we would like to compare how the theory-driven approach compares with data-mining approaches. In this section, we compare the results of the theory-driven approach to detect frustration, with some existing data-mining approaches, applied to the data from the Mindspark log file. However, given the limited information available from the Mindspark log data it is not easy to construct a Decision Network, as done in [33] and [139]. Hence, we consider the approaches used in AutoTutor [48], Crystal Island [111], and approach in [132], and apply them to the same Mindspark data.

We identified students' frustration using the data from the Mindspark log file by applying the approaches in [48],[111] and [132]. We identified 14 features from the Mindspark log file related to the students' responses, time spent to answer and time spent on reading explanation. We captured these features, after the students answered each question in the session. Features selected from the Mindspark log data with description are given in Table 5.5. According to the approaches in AutoTutor [48] and programming lab [132], we did a correlation analysis of these 14 features with the observed affective state to select those features that correlated with observed frustration. We identified 10 such features that were correlated with observed frustration, and omitted the remaining uncorrelated features. To avoid redundancy and to reduce the number of features, we did correlation analysis among features and removed the strongly correlated features (Pearson's r > 0.7) as suggested in [48]. If two features are highly correlated then the feature which has a higher correlation with the affective state is preserved [48].

Table 5.9: Description of Features Selected from Mindspark Log Data

Feature	**Description**
Ques No	Current question number in the session
Response	The response given by student is correct or not
Res Time	Time taken by the student to answer the question
Expln Time	Time spent on reading explanation after answering the question
Chall Res	The response by the student for the challenge question is correct or not
Prev Ques Res	The response to previous question
Pre Expln Time	Time spent on explanation of previous question
Last 2 Res	Sum of response to last two questions
Last 3 Res	Sum of response to last three questions
Last 4 Res	Sum of response to last four questions
Last 5 Res	Sum of response to last five questions
Last 6 Res	Sum of response to last six questions
Last 3 Res Time	Average response time for last three questions
Last 5 Res Time	Average response time for last five questions

After this analysis, seven features were selected: i) response to answer the question (whether the response provided by a student is correct or not), ii) response time to answer the question, iii) time spent on explanation of the answer, iv) response to answer the Challenge Question, v) sum of responses to the previous two questions, vi) sum of responses to the previous four questions and vii) average response time to answer the previous three questions. We used these seven features to repeat the analysis done in [48] and [132]. We applied the data to all the classifiers used in AutoTutor [48], to identify frustration from the log file. We report only the best result of the classifiers in each category in Table 5.10.

Similarly, we used all 14 features to repeat the analysis as done in Crystal Island [111], and we did experiments on Naive Bayes, SVM and Decision Tree classifiers as mentioned in [111]. The results, along with the theory-driven approach are shown in Table 5.10. We used the tenfold cross validation method in all our analyses.

Table 5.10: Comparison of our Approach with Existing Data-Mining Approaches Applied to the Data from Mindspark Log File

System	Classifiers	Accuracy in %	Precision in %	Recall in %
AutoTutor approach [48] (Selected Features)	Naive Bayes	82.83	40.94	37.95
	MLP	86.59	55.76	42.33
	K*	87.02	56.89	48.17
	Bagging Pred	87.55	57.89	56.20
	Logistic Model Tree	**88.63**	**65.97**	**46.71**
	PART	87.23	60.97	36.49
Crystal Island approach [111] (All Features)	Naive Bayes	81.12	38.72	48.90
	Decision Tree	**86.05**	**52.63**	**51.09**
Introductory programming lab approach [132]	Selected features used to form a linear regression model	r = 0.583		
Our Approach	**Linear Regression Model**	**88.84**	**78.94**	**32.85**

Bold – Best results obtained in each approach

In Table 5.10, the best results by each approach are highlighted. The detailed description of each approach is given here.

Autotutor Approach

We have identified seven features by correlation analysis as mentioned in Autotutor [48] from the Mindspark log data. We applied these seven features in weka [108] and used all classifiers mentioned in [48] to identify frustration from the log data. The classifiers used are given below with categories:

- Bayesian classifiers
  - Naive Bayes, Naive Bayes Updatable
- Functions
  - Logistic Regression, Multilayer Perceptron, Support Vector Machines
- Instance based techniques
  - Nearest Neighbor, K* (K Star), Locally Weighted Learning
- Meta classifiers
  - AdaBoost, Bagging Predictors, Additive Logistic Regression
- Tree based classifiers
  - C4.5, Decision Trees, Logistic Model Trees (LMT), REP Tree
- Rule based classifiers
  - Decision Tables, Nearest Neighbour Generalization, PART

In Table 5.10 we have shown only the best result of the classifiers in each category. Within the AutoTutor [48] approach, the Logistic Model tree performs comparatively better than the other classifiers on the Mindspark dataset with an accuracy of 88.63%, a precision of 65.97%, and a recall of 46.71%.

Crystal Island Approach

We have applied all the features identified from Mindspark log data in weka [108]. We have done experiments on Naive Bayes, SVM and Decision Tree classifiers as mentioned in [111]. We used tenfold cross validation to avoid bias in the training dataset. In the Crystal Island [111] approach, we observed that the Decision Tree classifier gives maximum accuracy of 86.05%, precision of 52.63%, and a recall of 51.09% as compared to other classifiers.

Programming Lab Approach

We used seven selected features to form a multiple linear regression model and tested the model using tenfold cross validation. Weka [108] is used to create and test the model as suggested in [132]. The results are correlation co-efficient $r = 0.583$.

From the Table 5.10 we observe that the accuracy and precision of our frustration model are high compared to the other approaches. However, the theory-driven approach performed poorly in recall of 32.85% (best result in data-mining approach is recall of 56.2%). As we mentioned earlier, we are interested in how correct our detection of frustration is (measured by Precision) instead of detecting all frustration instances encountered by students (reflected by Recall). Hence, our goal of achieving best precision and accuracy is achieved in the theory-driven approach. The reason for better precision and poor recall could be that the features are selected based only on goal-blockage type of frustration, and hence, other types of frustration might have been missed.

5.6 Performance of Theory-Driven Constructed Features Applied to Other Classifier Models

In our research, we started with the linear regression classifier model as our frustration model. To test, if different models for frustration (other than a linear model) perform better, we applied our features to non-linear models such as second order, and third order polynomial models. For the second order polynomial model, we combined the theory-driven features for the second order. Consider our features $goal1.bf$ as $f1$, $goal2.bf$ as $f2$, and so on, then our initial feature set will be $X = [f1, f2, f3, f4, f5]$. We combined these five features into second order features as $X2 = [f1f2, f1f3, f1f4, f1f5, f2f3, f2f4, f2f5, f3f4, f3f5, f4f5, f1f1, f2f2, f3f3, f4f4, f5f5, f1, f2, f3, f4, f5]$ and applied the linear regression algorithm to create the frustration model. The trained model is shown in Equation 5.13.

$$\begin{aligned} F_i = 0.8 * [-0.013 * f1 - 0.0984 * f2 + 0.0766 * f3 + 1.0002 * f4 - 0.0019 * f5 \\ -0.0984 * f1f2 + 0.0766 * f1f3 + 0.0292 * f1f4 + 1.7798 * f1f5 + 0.0766 * f2f3 \\ +0.0292 * f2f4 + 0.8444 * f2f5 + 0.0146 * f3f4 + 0.3325 * f4f5 + -0.0012 * f4f5 \\ + - 0.013 * f1f1 + 0.0363 * f2f2 + -0.1807 * f3f3 + 1.0002 * f4f4 \\ +0.0022 * f5f5 - 0.9997] + 0.2[F_{i-1}] \end{aligned} \quad (5.13)$$

From the above second order model, we observed that the weights of feature 5 (f5), which is time spent to answer the questions, is increased when combined with other features. This indicates the importance of time spent to achieve the goal in detecting frustration. For the third order polynomial model, the feature set will further add fifteen features to the second order feature set. To avoid handling the higher number of features, we used the polynomial kernel function in the support vector machine (SVM) to compute the performance of third and fourth order polynomial models. The polynomial kernel in SVM computes all the possible features based on the exponent (order) value given to it. The results of the second, the third and the fourth order polynomial models are shown in Table 5.11

From the results, we observe that the second and the third order polynomial model perform relatively better in precision, compared to the linear regression frustration model. However, the overall performance value (kappa) is comparatively lesser than the linear regression model. For the better understanding of cause of frustration, we use the linear regression model.

Table 5.11: Performance of Theory-Driven Features when Applied to Higher Order Polynomial Models

Order of Polynomial Model	Precision	Recall	Accuracy	Kappa
First	78.94%	32.85%	88.84%	0.41
Second	85.1%	29.2%	88.84%	0.3889
Third	82.4%	30.7%	88.84%	0.3989
Fourth	77.4%	29.9%	88.4%	0.3808

Table 5.12: Performance of Theory-Driven Features on Different Classifiers

Classifiers	Precision in %	Recall in %	Accuracy in %	Kappa
Naive Bayes	55.24%	57.66%	86.91%	0.4873
Logistic Regression	77.94%	38.69%	89.38%	0.4649
Bagging Pred	60.18%	49.64%	87.77%	0.4741
Logistic Model Tree	79.69%	37.23%	89.38%	0.4566
Decision Table	68.97%	43.80%	88.84%	0.4759

Further to measure the performance of theory-driven constructed features on non-linear classifiers, we applied our features to the non-linear classifiers in Weka[108]. The best result of the classifiers in each of the categories using tenfold cross validation method is shown in Table 5.12.

From the performance of theory-driven constructed features on non-linear classifiers, the Logistic regression model and the Logistic Model Tree outperformed our model in recall of 38.69%; the other metrics are performed equal to the linear regression frustration model. For a simpler model we considered linear regression model in our research. We note that logistic regression is the best method for our theory-driven approach. We use logistic regression model to generalize our theory-driven approach in Chapter 7.

5.7 Generalization of our Model Within Mindspark

We have collected human observation data of class six students from two schools, hence, the frustration model may not be used as it is for all age groups and all school curriculum. In this section we discuss the limitations of generalizing our frustration model. We have collected data from two schools–one from Mumbai, India and the other from

Ahmedabad, India. Both the schools are following the Indian Certificate of Secondary Education (ICSE)[§] curriculum. Hence, we cannot use our model to all curriculum across Indian Schools. In order to apply our model to students of different age and other school curriculum, revalidation of human observation is required.

We have collected the data from 27 students–14 male and 13 female. We restrict our generalization only to the following in Mindspark log data:

- ICSE school curriculum
- Different locations across India
- For both genders
- Age Group of 10-13

To verify the above claims we trained our model with the Ahmedabad school data and tested the model with the Mumbai school data. The results are given below. The confusion matrix is shown in Table 5.13

Table 5.13: Confusion Matrix of Mumbai School Data

		Human Observation	
		Frustrated	**Non-Frustrated**
Model	**Frustrated**	15	0
Data	**Non-Frustrated**	18	212

The proposed model classified the frustration in new school data with **Accuracy** of 92.65%, **Precision** of 100%. In the Ahmedabad school data the system identified 43 frustration instances from 687 observations, 43/687 = 6.2%. When we applied our trained model to data from a new school, we identified 15 instances of frustration out of 245 observations, 15/245 = 6.12%. It proves that our model can be generalized for the data from ICSE standard schools across Mindspark for the age group of 10-13.

[§] http://www.cisce.org/

5.8 Discussion

In this chapter, we proposed and validated a theory-driven approach to detect frustration of a student working with an ITS to understand the reasons for why the student is frustrated. We focused on how correctly the frustration instances are detected, rather than detecting all frustration instances encountered by the students. The performance of theory-driven approach compared with existing data mining approaches when applied to Mindspark log data. The results performed comparatively better in precision of 78.94% (best result from data-mining approach is precision of 65.97%) and comparatively equal in accuracy with 88.84% (best result from data-mining approach is accuracy of 88.63%). Hence, our goal of achieving best precision and accuracy is achieved in the theory-driven approach. However, the theory-driven approach performed poorly in recall of 32.85% (best result in data-mining approach is recall of 56.2%). The reason for better precision and poor recall could be that the features are selected based only on goal-blockage type of frustration, and hence, other types of frustration might have been missed. A significant advantage of the theory-driven approach is that the features identified provides the reasons for students' frustration and can be useful for informed responding. This knowledge can give information on which variables to control while responding to students' frustration.

The frustration model discussed in this Chapter is specific to Mindspark. In order to apply our frustration model to other systems, the theory-driven approach explained in Section 5.1 should be used. To apply our theory-driven approach to other systems, careful thought is required to operationalize the blocking factors of goals. The goals of the students when they interact with the system should be captured; this is a limitation in the scalability of our approach. The results of the theory-driven approach are dependent on how well the goals are captured and how well the blocking factors of the goals are operationalized. In this research we observed students from an homogeneous group. The human observation is valid for those students that belong to our group. In order to apply our approach to different group of students, revalidation of human observations is required for student from different group.

In next chapter, we describe the strategies to avoid the negative consequences of frustration in a timely manner, during the student's interaction with the ITS.

Chapter 6

Responding to Frustration

Affective computing is detecting and responding to the affective states of the user. Our research aim is to detect and respond to the frustration of the students when they interact with the ITS. In the previous chapter we described our theory-driven approach to detect frustration of the students while they interact with the Mindspark–a web-based math ITS. In this chapter, we describe our approach to respond to frustration, and the results of our approach.

We detect frustration using the features constructed from the Mindspark log data by applying theoretical guidelines. We respond to frustration using the messages which are created based on theoretical definitions.

6.1 Our Approach to Respond to Frustration

The steps of our approach to respond to frustration is shown in Figure 6.1.

Detect the students' frustration with its reasons when the students interact with Mindspark (Step 1). We described the Step 1 in Chapter 5. Create the motivational messages to respond to frustration when system detects frustration (Step 2). The messages are displayed to the students based on the reasons for frustration. The algorithm to display the messages is created in Step 3. Implement the approach to respond to frustration in Mindspark and collect students' log data for validation (Step 4). Impact of our approach to respond to frustration is analyzed in Step 5. The Step 2 of our approach is described

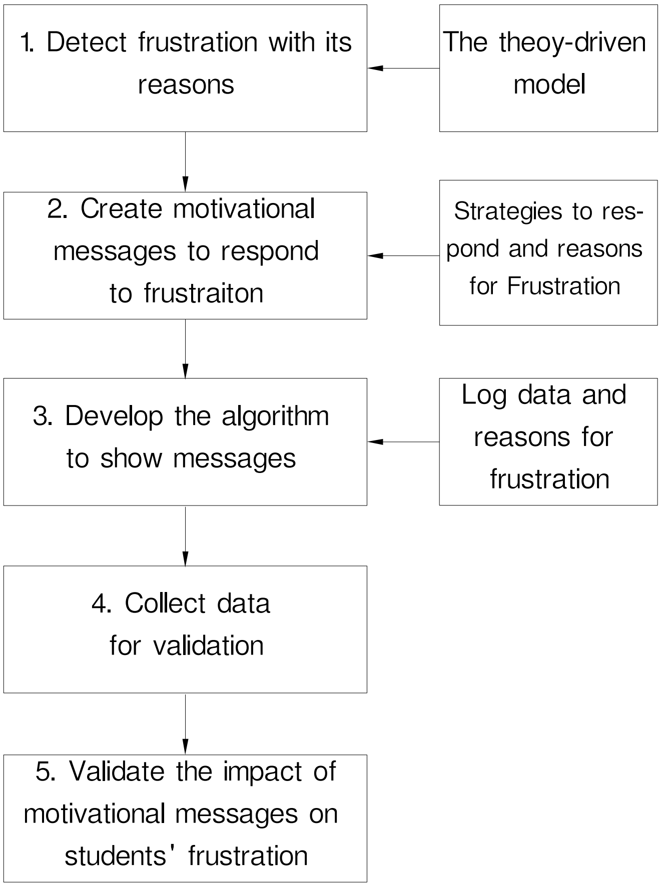

1. Detect frustration with its reasons
The theoy-driven model
2. Create motivational messages to respond to frustraiton
Strategies to res-pond and reasons for Frustration
3. Develop the algorithm to show messages
Log data and reasons for frustration
4. Collect data for validation
5. Validate the impact of motivational messages on students' frustration

Figure 6.1: Steps of our Approach to Respond to Frustration

in following subsections.

In Chapter 2, we describe in detail the theoretical basis of our strategy to respond to frustration. To summarize: the content in our motivational messages is based on attribution theory [160]. The attribution theory implies that motivating the students failure by messages which attributes the failure to external factors (such as math, difficulty of the question) will motivate them to set a new goal. We use Klein's guidelines [93], we provide the option to students to reflect their feedback, the feedback is requested after detecting frustration and feedback messages to show empathy for students' affective state. Using the recommendation from [129], [79] our motivational messages are displayed using agents who communicate empathy in their messages. Dweck's research [54] recommends that messages should be constructed to praise the effort not their intelligence.

Our strategy to respond to frustration consists of following aspects.

- Create motivational message to attribute the students' failure to achieve the goal to external factors [160].

- Create messages to praise the students' effort instead of outcome [54].

- Create messages with empathy, which should make the student feel that s/he is not alone in that affective state [93].

- Create message to request student's feedback [79].

- Display messages using an agent [129], [79].

6.1.1 Reasons for Frustration

We create and display the messages to motivate the students based on the reasons for why the student is frustrated. The prime reason for frustration is goal failure. The possible reasons for goal failure are identified from the students' goal while they interact with the ITS. We represent these reasons as "events." To create and display the messages we consider the events in Mindspark as listed in the Table 6.1. We modified our frustration model to identify the reasons for frustration (RoF) as shown in equation 6.1.

$$RoF = goal1.bf + goal2.bf + goal3.bf + goal4.bf \tag{6.1}$$

The values of RoF and its corresponding reasons for failure are detailed in Table 6.1. The value of RoF will be in the range of 0 to 5. If the $goal1$, that is getting correct answer to current question, is blocked then it is identified in $goal1.bf$ which is answer to current question is wrong. The current question can be a normal question or challenge question, which is identified the using the indicator I ($I = 0$ for normal question and $I = 1$ for Challenge Question).

Table 6.1: Explanation of the Reasons for Goal Failure - Events

Event	*RoF* Value	Answering Pattern
E1	0	The student's response to current question, a_i, is correct
E2	1	The student's response to current question, a_i, is wrong Normal and Challenge both
E3	2	The student's response to current question, a_i, is wrong and to previous question, a_{i-1}, was correct
E4	3	The student's response to current question, a_i, is wrong and to two previous questions, a_{i-2} and a_{i-1}, were correct
E5	4	The student's response to current question, a_i, is wrong and to three previous questions, a_{i-3}, a_{i-2} and a_{i-1}, were correct
E6	5	The student's response to current question, a_i, is wrong and to four previous questions, a_{i-4}, a_{i-3}, a_{i-2} and a_{i-1}, were correct

6.1.2 Motivational Messages

Our messages are created using the strategies discussed in previous section, the reasons for frustration and Mindspark log data. The following parameters are identified from the Mindspark log data, which are considered while creating the motivational messages. The messages are given in the Table 6.2 with condition to display the message and the reason for creating the message.

- Average Response Time (**Res Time**) is the average time taken to answer the questions in Mindspark by students. The average response time from Mindspark's existing log data, which we calculated, is 22 seconds.

- The Response Rate is the percentage of instances when students answered the question correctly. We calculate the response rate for all questions using the Mindspark's existing log data. We represent the response rate as **RR**. However, we found that RR of normal questions are higher then a Challenge question hence the RR is considered only in Challenge question.

- Frustration instances in the current session is represented by **FrusInst**. The FrusInst counts the number of frustration instances detected in the session.

The messages discussed in the Table 6.2 will be concatenated based on the conditions and displayed to the students. The message will appear as a speech bubble from a

Table 6.2: Messages to Respond to Frustration with Condition and Justification

	Condition to Display Message	**Message**	**Justification**
FrusInst = 1	E2 Challenge	You did well in the last four questions	The reason for frustration and to praise the student's effort
	E3	You did well in the last question	
	E4	You did well in the last two questions	
	E5	You did well in the last three questions and got a Sparkie!	
	E6	You did well in the last four questions	
	Res Time > Average response time	You tried hard to get the correct answer	Praising the students' effort.
	Res Time < Average response time	Try hard	To motivate the student
	Normal Question	I am sure you will do well in the next questions	
	Challenge Question	You may solve it next time.	
FrusInst = 2	Challenge Question	Dont worry, this is a tough question for many of your friends too. You can attempt it again.	Attributing the failure to difficulty of math question and motivating the student.
	Normal question and Response Rate > 50%	It is okay to get the wrong answer sometimes. You may have found the question hard, but practice will make it easier. Try again	To share the students feelings - show empathy
	Normal question and Response Rate < 50%	It seems this is a tough question for many of your friends too. Try again	Attributing the failure to difficulty of math question
FrusInst = 3	All questions	Would you like to give your feedback?	To receive student's feedback

buddy (an agent–shown in the Figure 6.3.) The algorithm to concatenate and display the messages is discussed next.

6.1.3 Algorithms to Display Motivational Messages

In this section, we describe step 3 of our approach, that is the algorithm to show motivational messages. For the events listed in Table 6.1, that is for each goal failure, we show the messages based on the student's response time in answering the questions from Mindspark, and nature of the question, that is how tough it is (how other students responded to the same question). We restrict the number of messages per Mindspark session to 3. This is to reduce the number of interventions to the students during their interaction with Mindspark. The messages are concatenated from the messages we created based on the conditions. We created algorithms to concatenate the messages from Table 6.2 for each event and described in this section in detail. We consider the following factors in all algorithms:

a) For the first instance of frustration, we choose the message based on the time spent by the student to answer the question, that is, Res Time. If the student spent more than an average response time then, based on the event, the message to praise the student's effort of answering the question will be shown. If the student spent less than an average response time then, the message to motivate the student will be shown. This is to praise the students effort to answer the question [54].

b) For the second instance of frustration, we choose the message based on the difficulty of the question, that is RR. If the RR is more than 50% then the message to motivate the student will be shown. If the RR is less than 50% then the message to attribute the failure to the difficulty of question will be shown. We assume that if the RR is $< 50\%$ then this question might be difficult for many of the students. This is to attribute the students' failure to difficulty of the question, hence student will be motivated for the next question [160].

c) For the third instance of frustration, the student's feedback is asked. The pseudo code for each algorithm is given below.

Event E1

The Student response to the current question is correct, so there will be no goal blockage and hence no motivational messages.

Event E2: The Student's Response to the Current Question is Wrong

If our model detects that the student is frustrated and the reason for frustration is that the student's response to the current questions is wrong, then the messages based on the following algorithm are shown. The current question can be a normal question or a challenge question. If the current question is a normal question, then the algorithm to show the message based on the instance of frustration in the current session is given in Algorithm 1.

Algorithm 1 To select the messages for Event 2 and Question Type is Normal

```
Require: Res Time, RR, FrusInst.
  return Message
  if FrusInst = 1 & Res Time > Average response time in seconds then
    Message: Good, you tried hard to get the correct answer. I am sure you
    will do well in the next questions.
  else if FrusInst = 1 & Res Time < Average response time in seconds then
    Message: Try hard. I am sure you will do well in the next questions.
  else if FrusInst = 2 & RR > 50% then
    Message: It is okay to get the wrong answer sometimes. You may have
    found the questions hard, but practice will make it easier. Try again.
  else if FrusInst = 2 & RR < 50% then
    Message It seems this question is tough, your friends also felt this question
    is tough, try again
  else if FrusInst = 3 then
    Message Would you like to give your feedback?
  end if
```

If the current question is a challenge question then student must have answered the last four questions correctly. So, the message to praise the student's effort on achieving the challenge question will be shown. Challenge questions are generally difficult questions, so, the RR of these questions is always less than 50%. If the current question is a challenge question, then the algorithm to show the message based on the instance of frustration in the current session is given in Algorithm 2.

Algorithm 2 To select the messages for Event 2 and Question Type is Challenge

```
Require: Res Time, RR, FrusInst.
  return Message
  if FrusInst = 1 & Res Time > Average response time in seconds then
    Message: You did well in the last four questions, and have tried hard to answer this challenge question too. Keep trying, you may solve it next time.
  else if FrusInst = 1 & Res Time < Average response time in seconds then
    Message: You did well in the last four questions. Try hard; you may solve it next time.
  else if FrusInst = 2 then
    Message Don't worry, this is a tough question for many of your friends too. You can attempt it again.
  else if FrusInst = 3 then
    Message Would you like to give your feedback?
  end if
```

Event E3: The Student's Response to the Current Question is Wrong and to the Previous Question was Correct

The messages based on the Algorithm 3 are shown if our model detects that the student is frustrated and the reason for frustration is that the student's response to the current question is wrong and to the previous question was correct. Since the student performed well in the previous question, the message praises the student's effort in the previous question.

Event E4: The Student's Response to the Current Question is Wrong and to the Two Previous Questions was Correct

If our model detects that the student is frustrated and the reason for frustration is that the student's response to the current question is wrong and to two previous questions was correct, then messages based on the Algorithm 4 are shown. Since the student performed well in two previous questions, the message praises the student's effort in the previous questions.

Algorithm 3 To select the messages for Event 3

```
Require: Res Time, RR, FrusInst.
  return Message
  if FrusInst = 1 & Res Time > Average response time in seconds then
    Message: You did well in the last question, and have tried hard to answer
    this question too. I am sure you will do well in the next questions.
  else if FrusInst = 1 & Res Time < Average response time in seconds then
    Message: You did well in the last question. Try hard; I am sure you will
    do well in the next questions.
  else if FrusInst = 2 & RR > 50% then
    Message: It is okay to get the wrong answer sometimes. You may have
    found the questions hard, but practice will make it easier. Try again.
  else if FrusInst = 2 & RR < 50% then
    Message It seems this question is tough, your friends also felt this question
    is tough, try again
  else if FrusInst = 3 then
    Message Would you like to give your feedback?
  end if
```

Algorithm 4 To select the messages for Event 4

```
Require: Res Time, RR, FrusInst.
  return Message
  if FrusInst = 1 & Res Time > Average response time in seconds then
    Message: You did well in the last two questions, and have tried hard to
    answer this question too. I am sure you will do well in the next questions.
  else if FrusInst = 1 & Res Time < Average response time in seconds then
    Message: You did well in the last two questions. Try hard; I am sure you
    will do well in the next questions.
  else if FrusInst = 2 & RR > 50% then
    Message: It is okay to get the wrong answer sometimes. You may have
    found the questions hard, but practice will make it easier. Try again.
  else if FrusInst = 2 & RR < 50% then
    Message It seems this is a tough question for many of your friends too. Try
    again. You may get a Sparkie next time.
  else if FrusInst = 3 then
    Message Would you like to give your feedback?
  end if
```

Event E5: The Student's Response to the Current Question is Wrong and to Three Previous Questions was Correct

If our model detects that the student is frustrated and the reason for frustration is that the student's response to the current question is wrong and to three previous questions was correct, then messages based on the Algorithm 5 are shown. Since the student performed well in three previous questions, the message praises the student's effort in the previous questions.

Algorithm 5 To select the messages for Event 5

```
Require: Res Time, RR, FrusInst.
  return Message
  if FrusInst = 1 & Res Time > Average response time in seconds then
    Message: You did well in the last three questions and got a Sparkie! You
    tried hard to answer this question too. I am sure you will do well in the
    next questions.
  else if FrusInst = 1 & Res Time < Average response time in seconds then
    Message: You did well in the last three questions. Try hard; I am sure
    you will do well in the next questions.
  else if FrusInst = 2 & RR > 50% then
    Message: It is okay to get the wrong answer sometimes. You may have
    found the questions hard, but practice will make it easier. Try again.
  else if FrusInst = 2 & RR < 50% then
    Message It seems this is a tough question for many of your friends too. Try
    again.
  else if FrusInst = 3 then
    Message Would you like to give your feedback?
  end if
```

Event E6: The Student's Response to the Current Question is Wrong and to Four Previous Questions was correct

If our model detects that the student is frustrated and the reason for frustration is that the student's response to the current question is wrong and to four previous questions was correct, then messages based on the Algorithm 6 are shown. Since the student performed well in four previous questions, the message praises the student's effort in the previous questions.

6.2 Methodology to Show Motivational Messages in Mindspark

The methodology to show the messages is explained in this section. The block diagram of the methodology to detect and respond to frustration is shown in Figure 6.2. As described

Algorithm 6 To select the messages for Event 6

```
Require: Res Time, RR, FrusInst.
  return Message
  if FrusInst = 1 & Res Time > Average response time in seconds then
    Message: You did well in the last four questions, and have tried hard to answer this question too. I am sure you will do well in the next questions.
  else if FrusInst = 1 & Res Time < Average response time in seconds then
    Message: You did well in the last four questions. Try hard; I am sure you will do well in the next questions.
  else if FrusInst = 2 & RR > 50% then
    Message: It is okay to get the wrong answer sometimes. You may have found the questions hard, but practice will make it easier. Try again.
  else if FrusInst = 2 & RR < 50% then
    Message It seems this is a tough question for many of your friends too. Try again. You may get a Challenge Question next time.
  else if FrusInst = 3 then
    Message Would you like to give your feedback?
  end if
```

in the previous section, the messages based on the event and the frustration instances in the current session are shown.

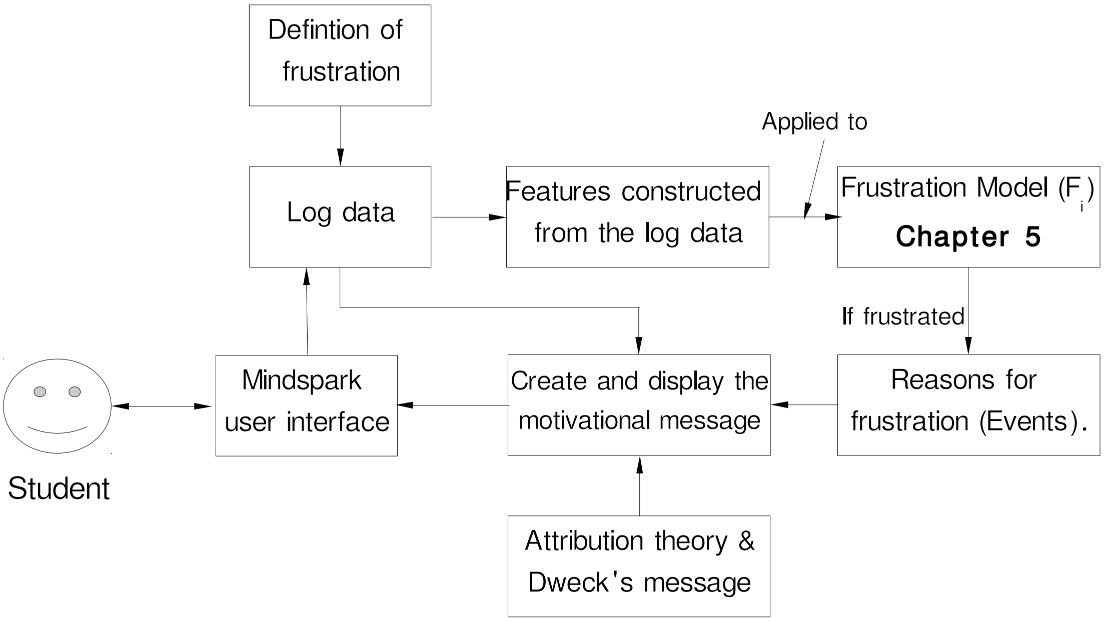

Defintion of frustration
Applied to
Log data
Features constructed from the log data
Frustration Model (F_i) Chapter 5
If frustrated
Student
Mindspark user interface
Create and display the motivational message
Reasons for frustration (Events).
Attribution theory & Dweck's message

Figure 6.2: Block Diagram of our Methodology to Detect and Respond to Frustration in Mindspark

As shown in Figure 6.2, the student's interactions with the Mindspark user interface are stored in the log file. From the log data the features to detect frustration are constructed using the definition of frustration. The constructed features are used to create the frustration model. The definition of frustration, the theory-driven approach to construct features from the log data, and validation of the frustration model are described in Chapter 5. If the student's frustration instances is detected by our frustration model, then the reasons for frustration are identified. The reasons for frustration are represented as events. The appropriate motivational message based on the events and the data from log file is selected, as explained in the Section 6.1.3. The implementation of our methodology in Mindspark is explained below.

6.2.1 Implementation of Affective Computing in Mindspark

In order to detect and respond to frustration in Mindspark, we integrated the frustration model and motivational messages in the Mindspark code. The following parameters are calculated from the Mindspark log data.

Data Required

- The student's response to Mindspark's questions (the result of the last five questions,) R in log data. We represent it as a_i.
- Time spent to answer the current question, S in log data.
- Question type, challenge or normal.

We use $R = a_i$ to represent the result to the current question; $a_i = 1$ if correct, $a_i = 0$ if wrong.
a_{i-1} is result of previous.
a_{i-2} is result of previous to previous question. And so on for, $a_{i-3}, a_{i-4}, a_{i-5}, ...$
Using the above data, we calculate following features:

- Feature 1: $goal1.bf(i) = (1 - a_i) * (1 - I)$

- Feature 2: $goal2.bf(i) = ((a_{i-2} * a_{i-1} * (1 - a_i)) + a_{i-1} * (1 - a_i)) * (1 - I)$

- Feature 3: $goal3.bf(i) = ((a_{i-4} * a_{i-3} * a_{i-2} * a_{i-1} * (1 - a_i)) + (a_{i-3} * a_{i-2} * a_{i-1} * (1 - a_i))) * (1 - I)$

- Feature 4: $goal4.bf(i) = I * (1 - a_i)$

- Feature 5: time spent to answer the question, $goal5.bf = S$

Where I is to check whether the current question is a challenge question or not. $I = 0$ for normal question and $I = 1$ for a challenge questio

The features are used to calculate the frustration index. The frustration model was trained using the data collected during human observation. The trained model from the equation 5.12 in Chapter 5 is reproduced below in the equation 6.2.

$$\begin{aligned} F_i = 0.8[0.147 + \; 0.423 \; * \; (Feature1 - 0.25) - \; 0.0301 \; * \; (Feature2 - 0.25)/2 \\ + \; 0.0115 \; * \; (Feature3 - 0.11)/2 \; + \; 0.8359 \; * \; (Feature4 - 0.04) \\ + \; 0.1864 \; * \; (S - 22.5)/300 \;] \; + \; 0.2 \; [F_{i-1}]. \end{aligned} \tag{6.2}$$

$F_i = 0$, for first three questions.

RoF = Feature1 + Feature2+ Feature3 + Feature4. (Range : 0 - 5)

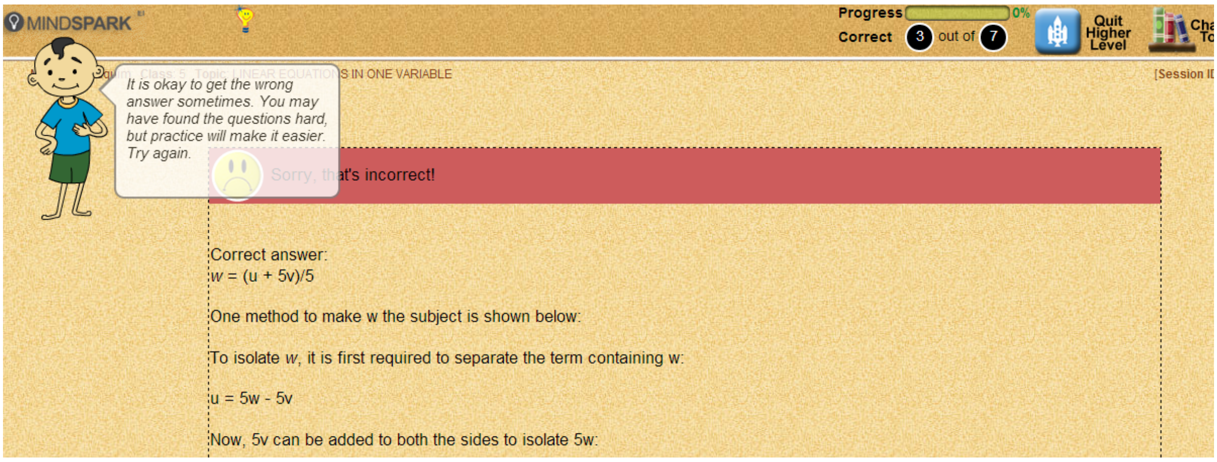
The algorithm to detect whether the student is frustrated or not, is given in Algorithm 7

The threshold value 0.5 chosen based on our analysis as shown in Table 5.6 in Chapter 5. From the threshold analysis, we observe that the kappa values are best for threshold values 0.2, 0.3, 0.4, and 0.5, however the precision value is very low compared to the threshold value = 0.5. The threshold value of 0.5 gives the best precision and recall combination.

Algorithm 7 To detect whether the student is frustrated or not

```
Require: $F_i$, $RoF$, FrusInst.
  return Message
  Initialize FrusInst = 0
  while After receiving response from student, $a_i$ do
    Calculate $F_i$ using $a_i$
    if The Frustration Index,$F_i$, is greater than the threshold value and FrusInst $<$ 3 then
      FrusInst++
      Based on the value of $RoF$, select the event and the algorithm to show motivational message
    end if
  end while
```

Motivational messages are shown to students in real-time when they interact with the Mindspark. The sample screen-shots of motivational messages shown to students is shown in Figure 6.3 and Figure 6.4.



It is okay to get the wrong answer sometimes. You may have found the questions hard, but practice will make it easier. Try again.

Sorry, that's incorrect!

Correct answer:
w = (u + 5v)/5

One method to make w the subject is shown below:

To isolate w, it is first required to separate the term containing w:

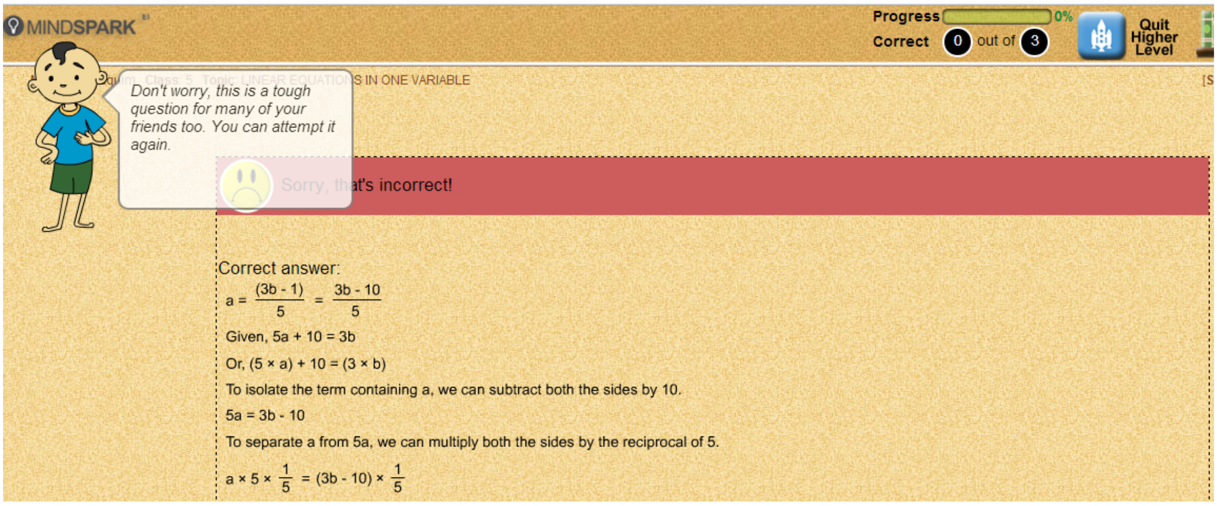
u = 5w - 5v

Now, 5v can be added to both the sides to isolate 5w:

Figure 6.3: Screen-Shot of Mindspark's Buddy Showing a Motivational Message as Speech Bubble - 1

6.3 Impact of Motivational Messages on Frustration

To determine the impact of the motivational messages, we compare the number of frustration instances without and with motivational messages in Mindspark sessions. In this section we discuss our data collection methodology, sample size and analysis techniques.



Don't worry, this is a tough question for many of your friends too. You can attempt it again.

Sorry, that's incorrect!

Correct answer:

$a = \frac{(3b - 1)}{5} = \frac{3b - 10}{5}$

Given, 5a + 10 = 3b

Or, (5 × a) + 10 = (3 × b)

To isolate the term containing a, we can subtract both the sides by 10.

5a = 3b - 10

To separate a from 5a, we can multiply both the sides by the reciprocal of 5.

$a \times 5 \times \frac{1}{5} = (3b - 10) \times \frac{1}{5}$

Figure 6.4: Screen-Shot of Mindspark's Buddy Showing a Motivational Message as Speech Bubble - 2

6.3.1 Methodology to Collect Data from Mindspark Sessions

The methodology to collect data to validate our approach is described in this section–step 4 of our approach to respond to frustration (Figure 6.1). Our methodology to collect the data is shown in Figure 6.5

6.3.2 Sample

We have created the frustration model based on the Indian Certificate of Secondary Education (ICSE*) syllabus. Hence, we test the impact of motivational messages in schools which follow ICSE syllabus. We collected the data from class six of three schools. Three schools were chosen based on the number of students using Mindspark in class six. Also we chose the school from different cities (Rajkot, Bangalore, Lucknow) in India.

6.3.3 Data Collection Procedure

The steps to collect data are given below:

- Extract Mindspark session data from one week of class six students in three schools.

*http://www.cisce.org/

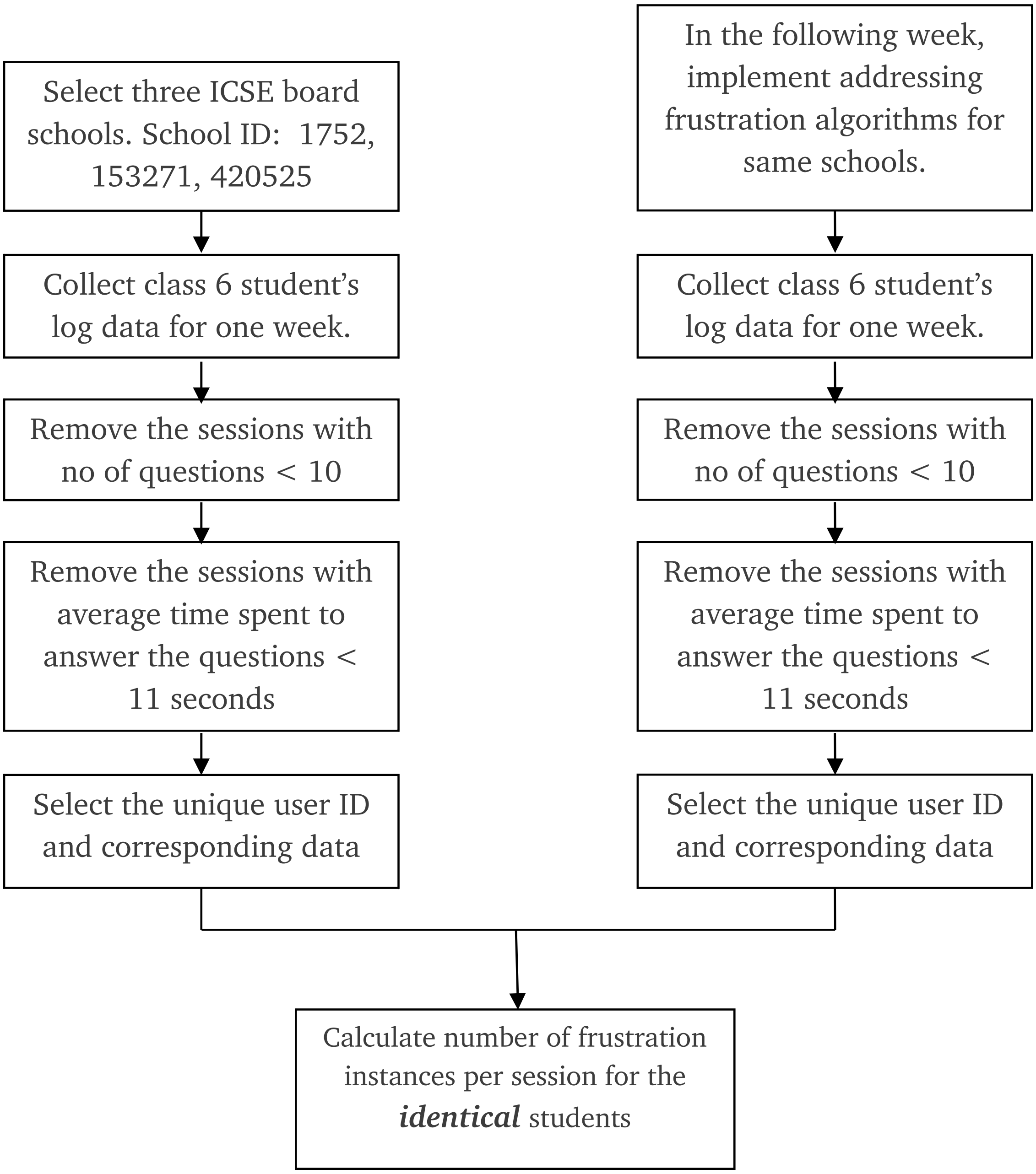

Select three ICSE board schools. School ID: 1752, 153271, 420525
Collect class 6 student's log data for one week.
Remove the sessions with no of questions < 10
Remove the sessions with average time spent to answer the questions < 11 seconds
Select the unique user ID and corresponding data
In the following week, implement addressing frustration algorithms for same schools.
Collect class 6 student's log data for one week.
Remove the sessions with no of questions < 10
Remove the sessions with average time spent to answer the questions < 11 seconds
Select the unique user ID and corresponding data
Calculate number of frustration instances per session for the ***identical*** students

Figure 6.5: Methodology to collect data for validating our approach to respond to frustration

- Remove sessions which have less than ten questions. This is to avoid having a small number of questions in a session.

- Remove those sessions which have less than 11 seconds as a average time spent on each questions.

- Retain the unique userIDs and their Mindspark sessions' data.

The selected student's userIDs were stored. In the following week, we implemented our approach to show motivational messages to the same schools. Motivational messages are shown to students through "buddy," based on the reasons for goal failure. These data collection steps are repeated to select the unique userIDs and their Mindspark sessions' data in the second week.

From the two weeks' data, the identical userIDs (students) and corresponding sessions' data are considered for analysis. We collected the Mindspark session data 188 students' from three schools for two weeks. The details of the data collected from the three schools are shown in Table 6.3.

Table 6.3: Details of the data collected from three schools to measure the impact of motivational messages on frustration

School Code	Number of students in Class 6	Mindspark topic in first week (Without motivational Messages)	Mindspark topic in second week (with motivational messages)	Number of matching students' sessions considered for analysis
1752	326	Integers	Integers	54
153271	279	Decimals	Decimals	72
420525	164	Algebra	Geometry	62
Total				188

6.3.4 Data Analysis Technique

The data from 188 students' Mindspark sessions are collected across two weeks has been used for our analysis. We used the trained frustration model, as shown in Equation 6.2 to detect the number of frustration instances in each session. The number of frustration instances in the first week's data from 188 students', without any motivational messages, was calculated. The number of frustration instances in the second week's data from 188 students', with motivational messages to avoid the negative consequences of frustration, were also calculated. The first Mindspark session of 188 students in each week is considered in our analysis, hence 188 Mindspark sessions data from two weeks are used in our analysis.

To analyze the impact of frustration, we compare the number of frustration instances in the data from the first and second weeks. The median is used as a metric to compare the frustration instances.

6.4 Results

The impact of motivational messages on frustration instances is discussed in this section, which is step 5 in the approach to respond to frustration. The median absolute deviation of the number of frustration instances in Mindspark sessions of 188 students is calculated and shown in Table 6.4.

Table 6.4: Median and Median Absolute Deviation (MAD) of number of frustration instances from the Mindspark session data from three schools

Number of Mindspark Sessions	Median of Frustration Instances	MAD of Frustration Instances
188 sessions without motivational messages	2	2.1942
188 sessions with motivational messages	1	1.4628

The number of frustration instances in 188 sessions are visually represented using the box plot in Figure 6.6. The figure shows the frustration instances without and with motivational messages.

The number of frustration instances after implementing motivational messages is reduced–shown in the box plot. The circle indicates out-lier; this means that the number of frustration instances per session being equal to six is very less after implementing motivational messages to respond to frustration. From the results, the median and median absolute deviation of frustration instances after implementing motivational messages to avoid frustration is reduced from 2 to 1 and 2.19 to 1.46, respectively.

To analyze the distribution of frustration instances, we plot the histogram of frustration instances without and with motivational messages. The histogram plot of the

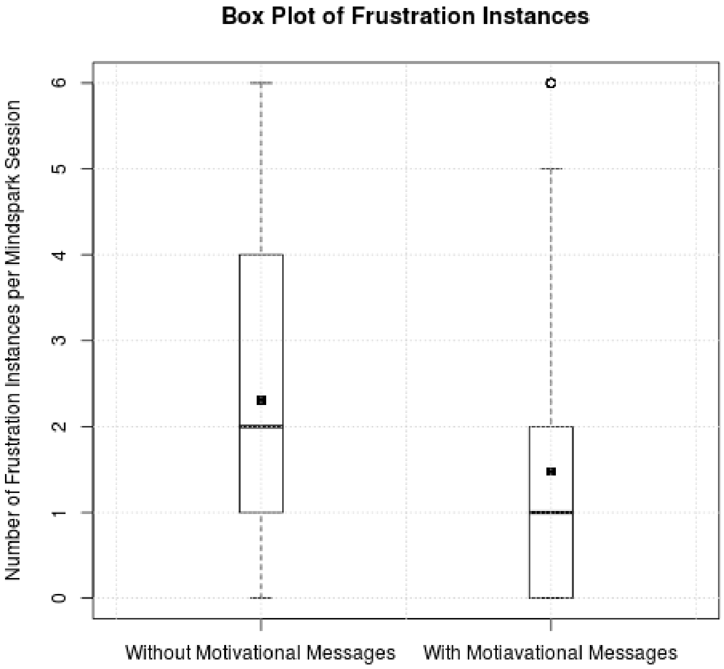
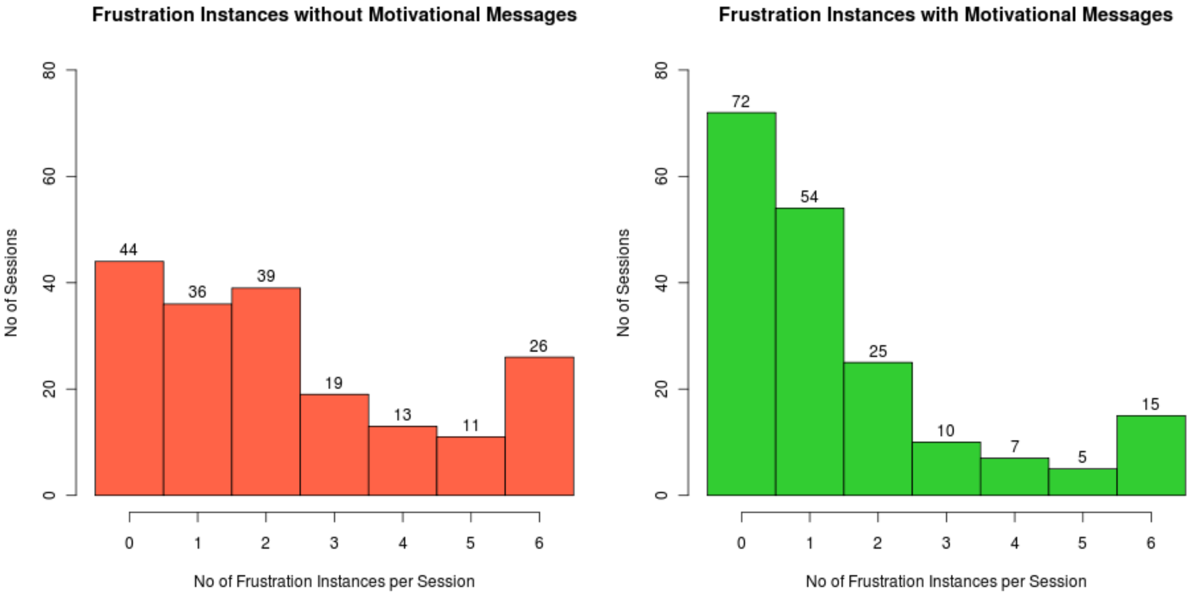


Figure 6.6: Box plot of Frustration instances from 188 sessions without and with motivational messages. Box = 25th and 75th percentiles; bars = minimum and maximum values; center line = median; and black dot = mean.

number of frustration instances without motivational messages is shown in Figure 6.7 (a). The histogram plot of the number of frustration instances with motivational messages is shown in Figure 6.7 (b).

The histogram plot shows that due to our approach, number of sessions (Y axis) with frustration instances per session (X axis) greater than or equal to 2 is reduced compared to the histogram plot without motivational messages. For example, number of sessions with frustration instances equal to 5 is reduced from 11 to 5 due to motivational messages. It indicates that the students' number of frustration instance per session has reduced. The histogram does not follow a normal distribution, hence, we used MannWhitney (MW) test to validate whether the two distributions are the same or not. The P value from MW test is = 0.0001 ($P < 0.05$), which rejects the null hypothesis that two distributions are same. The mean of ranks of frustration instances in sessions without motivational messages is high, compared to the sessions with motivational messages, 212.7



(a) (b)

Figure 6.7: (a). Histogram of Number of the Frustration Instances in 188 sessions without Motivational Messages and (b). Histogram of Number of the Frustration Instances in 188 sessions with Motivational Messages

vs 164.3. This indicates that the number of frustration instances are significantly lesser when motivational messages are shown to the students.

We further analyzed the significant difference in frustration instances for three schools separately, and the results of our analysis are shown in Table 6.5.

Table 6.5: Impact of motivational messages on frustration in three schools

School Code	Number of Sessions	Without Motivational Message		With Motivational Messages		Mann-Whitney's Significance Test
		Sum of Frustration instances	Median	Sum of Frustration instances	Median	
1752	54	92	1	57	0	$P < 0.05$
153271	72	212	3	148	1	$P < 0.05$
420525	62	130	2	72	1	$P < 0.05$

The P value from the MW test is = 0.0136 ($P < 0.05$) for school 1752, which rejects the null hypothesis that two distributions of frustration instances are same. Similarly the P value for school 153271 is 0.0028 and for school 420525 is 0.0062, which are lesser then 0.05. Hence, the data from all schools rejects the null hypothesis. Moreover, the data from Table 6.5 shows that, in all three schools, the number of frustration instances after implementing the approach to avoid frustration is reduced. Hence, the number of frustration instances is reduced statistically significantly in all three schools. The math topics used in Mindspark sessions are Integers, Decimals and Algebra. Hence, the results indicate that the motivational messages to respond to frustration are generalizable to all topics in math and across different schools in Mindspark.

6.4.1 Validation of Impact of Motivational Messages

In the previous subsection, the results indicate that the motivational messages reduced the number of frustration instances in Mindspark sessions. However, the results do not show whether the frustration instances reduced due to motivational messages or due to students' usage of Mindspark in the second week for same math topic. Hence, in this subsection we analyzed one school's data for two weeks without motivational messages.

For the validity analysis we used the data from school 1752. We considered the Mindspark sessions of 99 students for two weeks. The Mindspark sessions are selected as described in the previous section. The math topic in Mindspark sessions was Whole Number Concept. The students' data from the same school has been analyzed; it showed (in Table 6.5) a significant reduction in the number of frustration instances with motivational messages. The box plot of frustration instances in Mindspark sessions for two weeks without motivational messages is shown in Figure 6.8.

From the box plot above, the median value of number of frustration instances per session is reduced in second week from 2 to 1. However, the box size and mean value are not reduced and the maximum number frustration instances per session remain same. Hence, we used the Mann Whitney test, to check whether there is a significant difference in number of frustration instances from the Mindspark sessions. The details are given in the Table 6.6

From the Table 6.6, the P value from Mann-Whitney test is = 0.45 ($P > 0.05$)

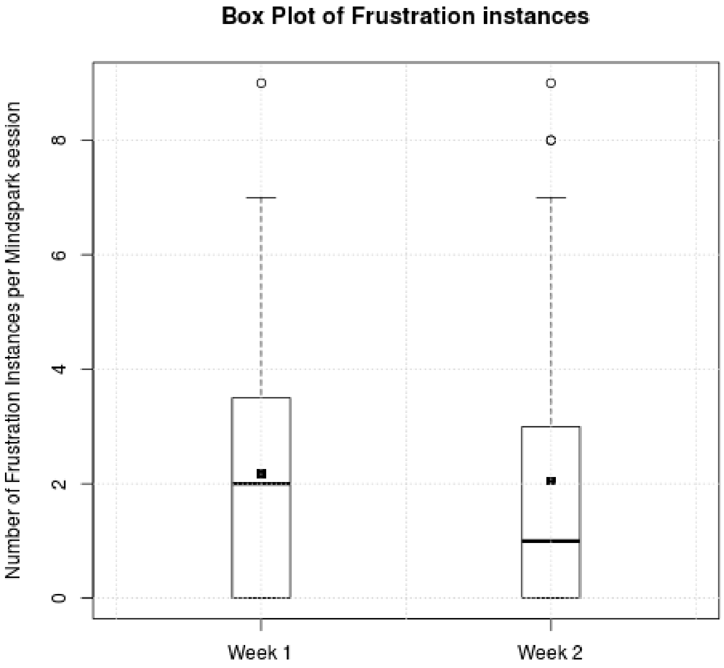


Figure 6.8: Box plot of Frustration instances from Mindspark sessions in a school for consecutive weeks without responding to frustration. Box = 25th and 75th percentiles; bars = minimum and maximum values; center line = median, and black dot = mean.

Table 6.6: Mann-Whitney significance test on frustration instances from Mindspark sessions without motivational messages

School Code	Number of Sessions	First Week Data		Second Week Data		Mann-Whitney's Significance Test
		Sum of Frustration instances	Median	Sum of Frustration instances	Median	
1752	99	215	2	203	1	P > 0.05

for school 1752, which accepts the null hypothesis that two distributions of frustration instances are same. This indicated that the number of frustration instances is not significantly reduced when the student uses the same math topic in Mindspark for two weeks. However, when the students' were learning same topic in Mindspark session showed significant reduction in number of frustration instances in second week with motivational messages. This has been established for same students (school 1752) in Table 6.5. Hence, we conclude that our previous results of significant reduction in frustration instances are due to the motivational messages shown to the students. To understand the impact of the motivational messages for students with different learning achievements, we analyzed our method in finer detail–described in the next subsection.

6.4.2 Detailed Analysis

The objective of the detailed analysis is to understand the impact of the motivational messages on students' frustration, learning achievement and time spent to answer the question. We start with the transition matrix of frustration instances for two weeks, without and with motivational messages. The matrix helps us to identify how the frustration instances of each student change after receiving motivational messages. Then, we analyze the impact of the motivational messages (to avoid students' frustration) with the learning achievement of the students and the average time spent to answer questions in the Mindspark session.

Motivational messages are shown only to those students who are frustrated. Hence, we show the transition matrix of sessions with frustration instances more than or equal to one. From 188 Mindspark sessions, we considered data from 88 Mindspark sessions which showed frustration instances at-least once in both weeks. Mindspark sessions of 88 students without motivational messages and the Mindspark sessions of the same students with motivational messages are considered to create a transition matrix.

The transition matrix of the Mindspark sessions of 88 students, for a number of frustration instances per session, from without motivational messages to with motivational messages is shown in the Table 6.7. The upper bound is considered as 6, that is the Mindspark session with more than 6 frustration instances is counted as 6.

The transition matrix Table 6.7, shows how the frustration instances in each student's Mindspark session change after receiving motivational messages. For example, consider the Mindspark sessions in the first week (without motivational messages) with number of frustration instances equal to two. From the table, it is 22. During the second

Table 6.7: Transition matrix of the Mindspark sessions of 88 students, for a number of frustration instances per session, from without motivational messages to with motivational messages

	Number of Students' Mindspark sessions with motivational messages							
	Frustration Instances per session	1	2	3	4	5	6	Total
Number of students' Mindspark sessions Without Motivational Messages	1	13	5		2		2	22
	2	12	5	3	2			22
	3	4	5	1		1	1	12
	4	2	1	2		1	3	9
	5	6					1	7
	6	4	3	1	1	3	4	16
	Total	41	19	7	5	5	11	88

week, with motivational messages, only 5 out 22 students got the number of frustration instances as equal to two. Also, the frustration instances of 12 out of 22 students is reduced to 1 and the remaining 5 students' per session is increased from 2 to 3 and 4. This indicates that the students' frustration instances reduce after receiving motivational messages. However, the students who got a lesser number of frustration instances per session without motivational messages are more frustrated in Mindspark sessions with motivational messages.

We compute the percentage of students' whose frustration instances are reduced due to motivational messages and the percentage of students whose frustration instances increased due to motivational messages. Table 6.8 shows the percentage of increase and decrease in frustration instances for 188 sessions and three schools separately. Moreover, we computed the percentage of frustration instances increased or decreased by greater than one steps, that is number of frustration instances per session change from 3 to 1, or from 4 to 0, or from 3 to 6, and so on. The percentage of "no change", that is, the students whose frustration instances are not changed due to motivational messages are not considered.

From the Table 6.8, it is evident that the percentage of reduction in frustration instances is almost double compared to the percentage of increase in frustration instances

Table 6.8: Percentage of increase and decrease in frustration instances at one and two steps. The data are from the Mindspark sessions of three schools. Percentage of no change is not shown

School	Number of Sessions	Frustration Instances Decreased		Frustration Instances Increased	
		By 1 step	By > 1 step	By 1 step	By > 1 step
Three schools	188	52.65%	33.51%	26.06%	13.82%
1752	54	44.44%	25.92%	20.37%	9.25%
153271	72	55.55%	34.72%	25%	16.66%
420525	62	53.22%	35.48%	32.25%	14.51%

due to motivational messages. Also, the frustration instances are reduced in all three schools. In school 1752, the percentage of frustration instances in week two increased less (9.25%) compared to school 153271 (16.66%). This can be due the math topic used by students of 1752 (same topic for two weeks). The frustration instances is reduced for all math topics, which infers that our motivational messages has impact on all math topics. However, the percentage of frustration decrease is varied for different math topic. We assume the is due to nature of the math topic considered.

6.4.2.1 Impact of Motivational Messages on Learning Achievement

The objective of this analysis is to check the impact of motivational messages on students with different learning achievement. We grouped the student ID's into low, medium, and high based on their performance in the Mindspark session. The students who got less than 60% of correct responses in the session are grouped as low, the students with 60% to 80% of correct responses in the session are grouped as medium, and the students with > 80% of correct responses in the session are grouped as high. Table 6.9 shows the impact of the motivational messages on the students with varying learning achievement.

From Table 6.9, the frustration instances of the students with low learning achievement are 77.78%, reduced compared to the 8.33% increase. The frustration instances of the students with medium learning achievement are 47.05% reduced compared to the

Table 6.9: Impact of the motivational messages on students with different learning achievement. The students are grouped based on their percentage of correct response in the Mindspark session. Low (< 60%), Medium (60% < and < 80%), and High (> 80%). The percentage of frustration instances which increased and decreased due to the motivational messages is shown. The percentage of no change in frustration instances are not shown

Learning Achievement	Number of Sessions	Frustration Instances Decreased	Frustration Instances Increased
Low	36	77.78%	8.33%
Medium	85	47.05%	29.41%
High	67	47.76%	31.34%

29.41% increase. Similarly, the frustration instances of the students with high learning achievement are 47.76% reduced compared to the 31.34% increase. Hence, the motivational messages have more impact on students whose correct response percentage is low.

6.4.2.2 Impact of Motivational Messages on Average Time to Answer the Questions in Mindspark Session

The objective of this analysis is to check the impact of motivational messages on the average time taken to answer the questions in the Mindspark session. We grouped the student ID's into "less", "average", and "more" based on average time they spent to answer the questions. The students who spent less then 20 seconds per question in the Mindspark session are grouped as "less", the students who spent between 20 to 30 seconds per question are grouped as "average", and the students who spent more then 30 seconds per question in the Mindspark session are grouped as "more". Table 6.10 shows the impact of motivational messages on the average time spent by the students to answer the questions in Mindspark session.

From the Table 6.10, the frustration instances of students who spent lesser time in answering the questions are 43.39% reduced compared to a 33.96% increase. The frustration instances of students who spent an average time to answer the questions are 49.2% reduced compared to 34.92% increase. However, the frustration instances students who spent more time in answering the questions are 63.88% reduced compared to a 12.5%

Table 6.10: Impact of the motivational messages on the average time spent by the students to answer the questions in the Mindspark session. The students are grouped based on their average time spent to answer the questions in the Mindspark session. Less ($<$ 20 seconds), Average (20 seconds $<$ and $<$ 30 seconds), and More ($>$ 30 seconds). The percentage of frustration instances which increased and decreased due to the motivational messages is shown. The percentage of no change in frustration instances are not shown

Average time spent to answer the questions in the Mindspark session	Number of Sessions	Frustration Instances Decreased	Frustration Instances Increased
Less	53	43.39%	33.96%
Average	63	49.2%	34.92%
More	72	63.88%	12.5%

increase. Hence, the motivational messages have more impact on students who spend average or more time in answering the questions in the Mindspark session.

6.4.2.3 Analysis on Ordering Effects - Removal of Motivational Messages

In our research, the sessions with motivational messages always followed by sessions with no motivational messages. Hence, to analyze the ordering effect we collected the data from sessions with no motivational messages followed by the sessions with motivational messages. We collected 42 students' Mindspark sessions' data from a school (153271) for three weeks. First week data are from sessions with no motivational messages, second week data are from sessions with motivational messages, and third week data are from the sessions with no motivational messages.

The number of frustration instances from 42 sessions in each week is visually represented using the box plot in Figure 6.9.

From the Figure 6.9, it is clear that frustration instances is higher in session with no motivational messages compared to the sessions with motivational messages. It is valid for the sessions with no motivational messages followed or preceded by session with motivational messages. This analysis proves that, there is no ordering effect in our approach to respond to frustration.

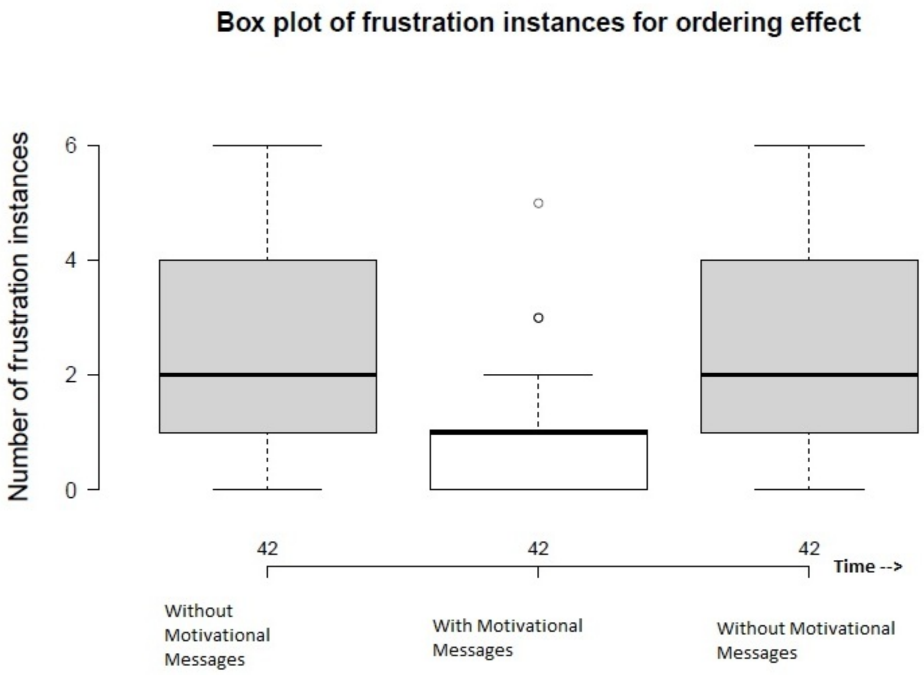


Figure 6.9: Box plot of Frustration instances from 42 session in each week. First week without motivational messages, second week with motivational messages and third week without motivational messages. Box = 25th and 75th percentiles; bars = minimum and maximum values; center line = median; and black dot = mean.

6.4.3 Discussion

In this chapter, we described our approach to respond to frustration in Mindspark. We analyzed the data from three schools to measure the impact of the approach to respond to frustration. The results are discussed above. From our analysis the following inferences are made.

- From the histograms in Figure 6.7, the frustration instances of students are reduced in the sessions with motivational messages.
- There is a statistically significant reduction in the number of frustration instances per session due to the approach to respond to frustration.
- The significant reduction in the frustration instances is independent of the schools analyzed and topics used in the Mindspark sessions.

- The approach to respond to frustration has a relatively higher impact on the students whose performance in the sessions is low.

- The approach to respond to frustration has a relatively higher impact on the students who spend more time to answer the questions in Mindspark session.

These inferences prove that, the approach to reduce frustration significantly reduces the number of frustration instances per session. Our research findings are aligned the existing research findings on responding to frustration. Scooter the tutor [133] reports that, due to motivational messages the number of frustration instances is reduced. Wayang Outpost [164] reports that, the students change their behavior because of motivational messages.

Our approach to respond to frustration can be generalized to all math topic and schools, in Mindspark. However, the impact of motivational messages are analyzed using students of class 6. The motivational message may have different impact on class 2 or class 3 students, who may not understand the meaning of the messages shown to them. Moreover, the higher class students might be self motivated and learn skills to handle frustration. Hence, this approach cannot be generalized to students of all ages.

Motivational messages, avoid the negative consequences of frustration and hence helps the students to continue learning and achieve their learning goal. Detecting and responding to frustration enriches the student model of ITS.

6.5 Summary

In this chapter we have discussed our approach to respond to frustration using the attribution theory. The approach we developed for Mindspark was based on students' goal failure and has been implemented and tested successfully. The results show that motivational messages using the attribution theory reduced the number of frustration instances per session. The reduction in the number of frustration instances is statistically significant. Also the approach can be generalizable to Math topics and class 6 students in schools, in

Mindspark.

Chapter 7

Generalizing Theory-Driven Approach

In this chapter, we motivate generalization of our theory-driven approach to detect other cognitive affective states. We discussed modeling frustration in Chapter 5. We now describe modeling "boredom" using the theory-driven method. Boredom is one of the cognitive affective state considered in the computer learning environments [26], that has been studied with respect student learning in ITS [39]. Our goal in this chapter is to show proof of concept of how the theory-driven approach we developed can be used to model the other affective states. We do not attempt to create a state of art model to detect boredom in ITS.

7.1 Applying Theory-Driven Approach to Model Boredom

In chapter 5, we described our theory-driven approach to model frustration in an ITS. The generic theory-driven approach to detect affective states is given below:

1. Operationalize the theoretical definition of affective state for the system under consideration.

2. Construct features from the system's log data; based on the theoretical definition of affective state.

3. Create a logistic regression model using the constructed features to detect the affective state.

4. Conduct an independent method to detect affective state and use the data from independent method to train the weights of logistic regression model.

5. Validate the performance of the model by detecting the affective state in the test data and compare the results with the data from independent method.

In this chapter, we modified the generic theory-driven approach to model boredom and is shown in Figure 7.1.

7.2 Definition of Boredom

In this is section we discuss the various existing definitions of boredom. Then we discuss the definition of boredom within educational settings.

In 1938, Barmack [13] defines boredom as a conflict between the tendency to continue the situation and tendency to escape from it due to lack of motivation. For example, if a student is not motivated to complete a assignment given to him/her, and if s/he is not interested in the subject, then the conflict arises whether to complete and submit the assignment or not to submit. That state of conflict is called as boredom. It is the first published scientific study on boredom.

In 1981, the review article by Smith [148], reviewed the psychological and psychiatric study of boredom from 1926 to 1979. The review article reports that, the common factors to generate boredom are repetitiveness, lack of novelty, and monotony.

In another review article [119], Hanlon reviews the research articles from 1930-1980 and lists the general agreement regarding boredom. They are:

- Boredom is a reaction to the repeated activity and monotonous stimulation.

- Degrees of boredom vary greatly for different individuals in the same working environment.

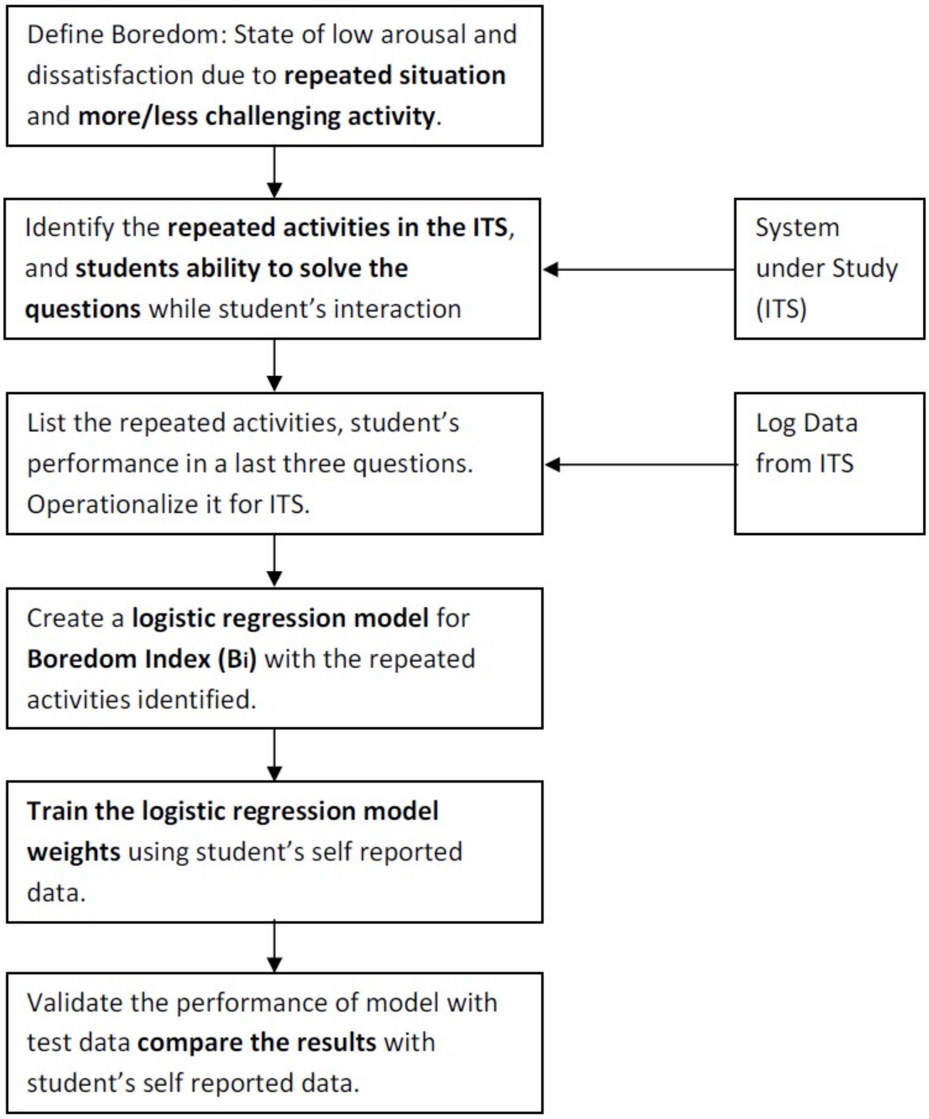

Define Boredom: State of low arousal and dissatisfaction due to **repeated situation** and **more/less challenging activity**.
Identify the **repeated activities in the ITS**, and **students ability to solve the questions** while student's interaction
System under Study (ITS)
List the repeated activities, student's performance in a last three questions. Operationalize it for ITS.
Log Data from ITS
Create a **logistic regression model** for **Boredom Index (Bi)** with the repeated activities identified.
Train the logistic regression model weights using student's self reported data.
Validate the performance of model with test data **compare the results** with student's self reported data.

Figure 7.1: Steps of theory-driven approach to create a boredom model using data from the Mindspark log file

- The person can escape boredom by changing the situation or escaping from the situation.

- Boredom is situation specific and it is reversible if the situation changes.

More recently, Mikulas and Vodanovich (1993), [114] defines boredom as "a state of relatively low arousal and dissatisfaction, which is attributed to an inadequately stimulating situation [114]." For example, boredom occurs if the student is in "non interesting" or "non challenging" situation which fails to keep his/her motivation. Flow theory (1997) by

Csikszentmihalyi [37] states that boredom occurs when the students' skill and challenge mismatches (high skill, less challenge). For example, if the student is asked to answer less challenging questions compared his skill then s/he might get bored. Carriere, Cheyne, and Smilek (2008), [28] states that boredom occurs due to the inability to engage or sustain attention in the current situation. Boredom is outcome of (a) forcing not to do the desirable action or (b) forced to do undesirable action.

Another recent article [158] in Educational Psychology Review journal, reviewed the definitions of boredom in educational settings. The article proposed a definition of boredom from their review of definitions of boredom in educational setting. "State boredom occurs when an individual experiences both the (objective) neurological state of low arousal and the (subjective) psychological state of dissatisfaction, frustration, or disinterest in response to the low arousal." This definition states that both the state of low arousal, that is less interest to the outcome of an event, and dissatisfaction with the outcome of an event leads to boredom.

7.2.1 Definition of Boredom Used in Our Research

From the above overview of boredom, the most common feature in all existing work on boredom is repetitiveness and monotonous stimulation [148], [119]. The other key features of boredom are

1. Conflict between whether to continue the current situation or not due to lack of motivation [13].

2. The student is forced to do the an uninteresting activity. Non-interest occurs when the student not challenged enough [114], [37].

3. The student is prevented from doing a desirable action or forced to do an undesirable action [119].

4. The student lost the interest in outcome of the event [158].

Out of these above features Mindspark captures log data related only to repetitiveness, less challenging activity and students' non-interest to the outcome of the event. Hence we consider only these features to model boredom.

7.3 Modeling Boredom

Guided by the theoretical definition (Step 1), we construct features from the ITS log data (Step 2). Based on information from the student log, the the features are constructed $f1, f2, ..fn$ (Step 3). We formulate a logistic regression model for B_i, the boredom index at i^{th} question based on the constructed features (Step 4). The weights of the logistic regression classifier are determined during training process (Step 5) with labeled affective states from student's self reported data – an independent method to identify the affective states. The performance of our model is validated by detecting boredom in the test data and comparing the results with the student's self reported data (Step 6).

The logistic regression model to detect boredom is given below:

$$B_i = w_0 + w_1 * f1 + w_2 * f2 + w_3 * f3 + ... + w_n * fn \tag{7.1}$$

The weights, $w_0, w_1, w_2, ..., w_n$, in the equation above are determined by the logistic regression analysis.

7.3.1 Boredom Model for Mindspark

The definitions of boredom are operationalized to construct features from the Mindspark log data. In this section we describe how we operationalize the definition of boredom for Mindspark log data and construct the features. The logistic regression model is used to detect student's boredom while they interact with the Mindspark.

We created the logistic regression model based on steps given in Figure 7.1.

Step 1. Boredom: We consider that boredom occurs due to following features: a) repeated activities [148], [119], b) disinterest in outcome of an event [158], and c) less challenging activity [114], [37].

Step 2. Operationalize: We operationalize the following from the Mindspark log data:

- Repeated activities: In Mindspark a student can get repeated activities in two following situations

  If the student fails in the topic then s/he will get the same identical question.

  If the student fails in the subtopic then s/he will get the similar questions with different numerical values in the question.

- Disinterest: The student interest to the outcome considered to be less, if the student spent less time to answer the question and less time to read the explanation if the question is wrongly answered. Hence we consider the response to the current question, the time taken to answer the question and time spent to read the explanation in the last question are considered.

- Less challenging activity: If the student is answering the questions correctly without much effort (less time) then we consider that the student is not challenged enough. In Mindspark we consider the student's performance in last three questions and the average time taken to answer the last three questions. These parameters will determine whether the student is challenged less or not.

Step 3. Boredom Variables: We construct the following features from Mindspark to model Boredom: repeating the same questions $Ra1$, repeating similar questions $Ra2$, the average time spent to respond to the last three questions At, the students' response in the last three questions, a_i, a_{i-1}, a_{i-3}, the time spent to answer the current question T_i, and the time spent to read the explanation of the last question ET_{i-1}, are constructed from the student's log data. i denotes the current question.

Step 4. Logistic Regression Model: The mathematical model to detect boredom for Mindspark data is given in Equation 7.2.

$$\begin{aligned} B_i = w_0 + w_1 * Ra1 + w_2 * Ra2 + w_3 * At + w_4 * a_i + \\ + w_5 * a_{i-1} + w_6 * a_{i-2} + w_7 * T_i + w_8 * ET_{i-1} \end{aligned} \tag{7.2}$$

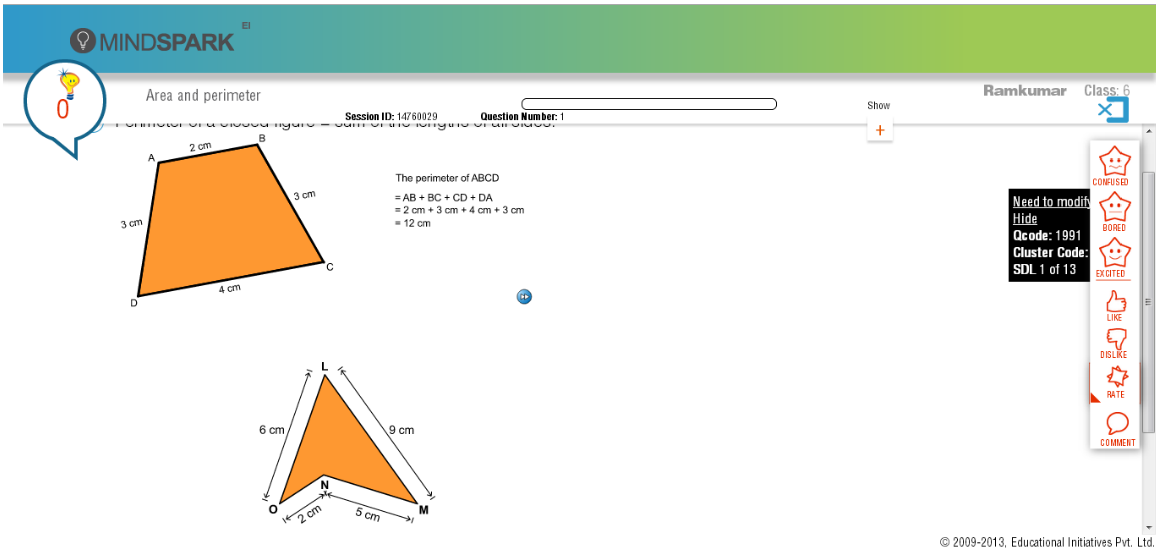
Step 5 and Step 6 of Figure 7.1 are explained in the following sections.

7.4 Experiment Methodology

In this section, we discuss the independent method we used to identify boredom: the student's self reported data. We describe sample data, and the metrics used to compare the results of our boredom in this section.

7.4.1 Self Reporting

Mindspark has an emotToolbar integrated with its user interface. It is optional and can be activated by mouse-over operation. The student can choose to report his/her emotions at any time. Minspark will not force the students to report their emotions. The emotToolbar is shown in Figure 7.2.



Session ID: 14760029 **Question Number:** 1

The perimeter of ABCD
= AB + BC + CD + DA
= 2 cm + 3 cm + 4 cm + 3 cm
= 12 cm

Qcode: 1991
Cluster Code:
SDL 1 of 13

© 2009-2013, Educational Initiatives Pvt. Ltd.

Figure 7.2: EmotToolbar integrated with Mindspark user interface to collect students' emotions. The emote bar is in right side of the figure.

The emotToolbar consists of six options for the students to choose from as shown in the Figure 7.3.

- Confused
- Bored
- Excited
- Like
- Dislike
- Option to give comments.

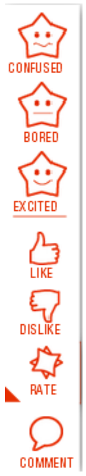


Figure 7.3: The EmotToolbar

The emotToolbar takes only one option at a time as an input from students, for example a student cannot choose Bored and Confused at the same time.

Sample

The student's self reporting on emotToolbar is optional in Mindspark, hence the question not marked as bored should not be considered as non-bored if there is no input from the student in the emotToolbar. However, to train a logistics regression classifier, both bored and non-bored data are required to create a model.

To address the issue mentioned above, we choose only those sessions which have five or six entries in the emotToolbar. The students who report their emotions (confused, bored, excited), for two or three times per session may not be reporting all the emotions they had undergone during their interaction with Mindspark session. Hence, we consider sessions where the students reported their emotions for five or six times. We collected 90 sessions data, which have reported emotions five or six times, from the Mindspark database. The sessions include different school boards (ICSE, CBSE) and different schools across India.

7.4.2 Analysis Procedure

We collected 1617 instances of student's answering the questions in Mindspark. Out of 1617, 442 instances are self reported as boredom (Bored) by students, the remaining instances are marked as (Non-Bored). The dataset is stratified at questions (instances) level. Hence, our unit of analysis is the instances where students respond to questions in Mindspark. We represent the values obtained from the emotToolbar as Bor_i at the i^{th} instance, $Bor_i = 0$ for non-bored and $Bor_i = 1$ for bored.

To create a boredom model using the logistic regression classifier (**Step 5** in Figure 7.1), we followed the similar steps that were used to model frustration in Chapter 5. The modified steps to model boredom are given below:

1. To maintain a uniform scale among all the features in the boredom model, we apply normalization to all the features. We used the following normalization equation:

$$X_{new} = \frac{X - Mean(X)}{Max(X) - Min(X)}$$

   Here, X_{new} is the normalized value of feature X.

2. We use cross-validation technique [66] in our analysis, to train and validate our model. In this research, we used the tenfold cross-validation technique [66] for all the experiments.

3. We use logistic regression analysis to identify the values for weight, by assigning 0 and 1 to represent non-bored and bored, respectively, in training dataset.

4. The trained model is used to detect boredom on test dataset. Validate our detection by comparing with student's self reported data.

7.4.3 Metrics to Validate Boredom Model

We use the same metrics used to measure the performance of our frustration model to measure the performance of the boredom model. However, we briefly discuss the metrics used in this section. The most common metric used in the classification problem is accuracy [113], precision and recall.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{7.3}$$

Accuracy measures the ratio of correctly detected instances (TP + TN) to total instances.

$$Precision = \frac{TP}{TP + FP} \tag{7.4}$$

Precision measures the ratio of correctly detected boredom instances (TP) to total number of detected boredom instances (TP + FP).

$$Recall = \frac{TP}{TP + FN} \tag{7.5}$$

Recall measures the ratio of correctly detected boredom instances (TP) to the actual number of boredom instances self reported by the students (TP + FN).

Since our dataset is an unbalanced distribution of bored and non-bored, we calculate the F1 score and Cohen's kappa to measure the performance of our model compared to random guesses.

7.4.4 Related Works - Detecting Boredom

We discuss the related works in detecting boredom using the data students' interaction. We discuss three such systems in this section: AutoTutor [48], Crystal Island [139] and Cognitive Tutor Algebra I [11]. In AutoTutor, boredom was detected using the data from log file and also using the data from external sensors such as body posture, and eye tracking [44]. In this section we discuss only the results of boredom detection using the data from log file.

AutoTutor [48] is a dialogue-based tutoring system. The affective states detected are frustration, boredom, flow and confusion. The features are identified based on the features from the log file, such as, response time, number of characters in a student's response, number of words in student's response, change in student's score and tutor feedback to a student's response. The data from 28 college students' interaction with AutoTutor were considered for the analysis. Independent methods used in this research to detect affective states and validate the results were self reporting, peer reporting and human observation of students' facial expressions. Affective states were detected for every 20-second interval and the results are compared with the affective states from independent method. This study reports individual detection accuracy of boredom, confusion, flow, and frustration, when compared with neutral affect are 69%, 68%, 71%, and 78%, respectively. The detection accuracy for boredom is 69%.

Affective state modeling in Crystal Island [139], creates a Dynamic Bayesian Network (DBN) model to capture the students' affective states. The affective states detected in this system was anxious, bored, confused, curious, excited, focused, and frustrated. The features considered to model students affective states are personal attributes of students which are identified from students' scores and personal surveys prior to interaction with system, observable environment variables such as goals completed, books viewed and successful tests, and appraisal variables which are student's cognitive appraisals such as learning focus and performance focus. The data from 260 students' interactions with Crystal Island was considered to model the affective states. Naive Bayes, Bayes net and dynamic Bayesian net were used to model the affective states. The independent method used in this research to validate the results is self reporting. The student's were asked to self-report their current mood in the dialog box for every seven minutes. The students were asked to select the emotion from set of seven emotions which are anxious, bored, confused, curious, excited, focused, and frustrated. The individual detection accuracy reported for anxious, bored, confused, curious, excited, focused, and frustrated are 2%, 18%, 32%, 38%, 19%, 52% and 28%, respectively. The detection accuracy for boredom is 18%.

Cognitive Tutor Algebra I [11] is a algebra tutor used is schools as part of regular mathematics curriculum. The affective states detected in this system are boredom, confusion, engaged concentration, and frustration. Total of 58 features were created from the students' behavior during interaction with system and past performance. The features are collected for every 20 second window. The features like student response to the current question, help request, and previous performance in current skill were considered to detect the affective states. The data from 89 students' interaction with the Cognitive Tutor Algebra I are considered for the analysis. Independent method used in this research to detect affective states and validate the results was human observation of students' facial expressions. Affective states were detected for every 20-second interval and the results are compared with the affective states from independent method. The Kappa score was used to measure the goodness of detection model for each affective states. The reported Kappa scores were 0.31 for engaged concentration, 0.40 for confusion, 0.23 for frustration, and 0.28 for boredom. For boredom, the best algorithm reported was Nave Bayes with Kappa = 0.28.

From the above related works the detection accuracy of boredom is comparatively lesser than detection accuracy of frustration. The best detection accuracy of boredom is 69% as reported in [48]. The best reported Kappa value of boredom model is 0.28 [11].

7.5 Results

In this section we describe the performance our boredom model. The extracted features are applied to weka [108]. The logistic regression model to classify boredom is:

$$\begin{aligned} B_i = -0.66 + 0.05 * Ra1 - 0.06 * Ra2 + 0.94 * At - 0.48 * a_i + \\ -0.27 * a_{i-1} + 0.314 * a_{i-2} + 7.6 * T_i - 2.73 * ET_{i-1} \end{aligned} \tag{7.6}$$

The performance of our boredom model on the Mindspark data, using tenfold cross-validation, compared to self reported observation is given in Table 7.1.

Table 7.1: Contingency Table of Our Approach when Applied to Mindspark Log Data

		Self Reported Data	
		Bored	**Non-Bored**
Pred	**Bored**	98	46
Result	**Non-Bored**	344	1129

The values from Table 7.1 are used to calculate the performance of our model. The results are given in Table 7.2.

Table 7.2: Performance of our Approach Shown Using Various Metrics when Applied to Mindspark Log Data

Metrics	**Results**
Accuracy	75.88%
Precision	68.1%
Recall	22.22%
Cohen's kappa	0.23
F1 Score	0.33

From the results, the accuracy and precision of our boredom model are high, compared to recall.

7.5.1 Discussion

Our boredom model performed relatively better compared to the existing research work in accuracy with 75.88%. Precision and recall are not reported in the existing research work. The value of F1 score and Kappa in our approach indicates that our model can be improved.

Moreover, compared to our frustration model, boredom model performed poorly in accuracy with 75% vs 88%, in precision 68% vs 79% and in recall 22% vs 32%. The reason for poor performance can be attributed to consideration of few factors of boredom in operationalization and validity of self reporting. The students' age is in the range of 10-12, due to this, the students might have wrongly reported other emotion as boredom or vice versa. The advantage of our model is that the features identified gives the reasons for students' boredom and it is useful while addressing the student's boredom.

7.6 Conclusion

To check the generalization of our theory-driven approach to model other affective states, in this chapter, we applied our theory-driven approach to detect boredom of the students while working with an ITS. The process shows that our theory-driven approach is generalizable to model other affective states and not only frustration. However, the performance of the model is lesser compared to the theory-driven frustration model. The reason for the lower precision, recall and kappa can attributed to the operationalization of features from the Mindspark log data and independent method used to detect boredom. Improving the operationalization of the theory for the ITS, and independent method such as human observation might improve the performance of the boredom model. However, our main goal of theory-driven approach, which is understanding the reasons for boredom, is shown in our model. Our boredom model gives the reasons for students' boredom and it can be useful to respond to boredom.

Chapter 8

Summary and Conclusion

In this chapter we summarize our research work with results, contributions and describe possible future works from our research.

8.1 Summary

In this research work, we propose to detect and respond to students' frustration while working with an ITS. We developed a frustration model using the features constructed from the log data by applying theoretical definitions of frustration. We implemented our model in an ITS, Mindspark. By modeling frustration using theoretical definitions, the reasons for frustration can be inferred. Since the reasons for frustration is known, it led to informed responding to frustration. The data from independent method (human observation) was used to create and validate the frustration model. The results are described in Chapter 5.

In our research, what is important is how accurately the frustration instances are detected rather than detecting all frustration instances encountered by the students. Our theory-driven approach performed comparatively better than other approaches, with a precision of 78.94% (the best result from the data-mining approach with a precision of 65.97%) and comparatively equal in accuracy with 88.84% (the best result from the data-mining approach is with an accuracy of 88.63%). Hence, our goal of achieving the best precision is achieved in our theory-driven approach. However, the theory-driven approach performed poorly in recall of 32.85% (the best result from the data-mining approach with a recall of 56.2%). The reason for better precision and poor recall could be that our frustration model detected the students' frustration which occurred only due to interaction with Mindspark not the frustrations which occurred due to external factors such as Internet speed and interaction with friends.

In order to apply our frustration model to other systems, careful thought is required to operationalize the factors that block the goals. Moreover, the goals of the students when they interact with the system should be determined. These are the limitations in the scalability of our model. The results of the frustration model are dependent on how well the goals are captured and how well the blocking factors of the goals are operationalized. Another limitation of our model is, we used the data from class six students from two schools. Hence, our model can be generalized only for the students with the age group of 10–13, across Mindspark.

Our theory-driven approach can be generalized to detect other affective states. In Chapter 7, we applied our theory-driven approach to detect boredom of students while they interact with the ITS. The results suggest that our model can be generalized to detect other affective states in ITS. The results of detecting boredom using our theory-driven approach in chapter 7, are comparatively equal to the existing methods. However, the kappa value (k = 0.23) indicates poor performance of our approach when applied to detect boredom. The reason for the lower kappa value can be attributed to the theories of boredom considered to operationalize the features from the Mindspark log data and the independent method used to detect boredom. More research on improving the operationalization of the theory for the ITS and independent methods, such as human observation, will improve the performance of the boredom model. Our boredom model gives an interpretation of the reasons for students' boredom and it can be useful to provide informed responses.

The aim of our research is to avoid the negative consequences of frustration (detected by our model) in real-time, at Mindspark's production environment. To achieve our aim, we integrated our model with the Mindspark code to check the students' frustration index after every response and trigger an appropriate algorithm whenever the frustration index is above the threshold value (0.5). We developed the algorithms to respond to the frustrations in a timely manner using motivational messages, during the students' interaction with the ITS. The algorithms are developed based on the reasons for frustration, the time taken to answer the question, the difficulty level of the question and the number of frustration instance in that session by the student.

The results of our strategies to respond to frustration are described in Chapter 6. The results show that the number of frustration instances is reduced due to motivational messages. The reduction in the number of frustration instances per session is statistically significant. Our approach can be generalized to schools across Mindspark and it is independent of schools analyzed and Mindspark topics used in the Mindspark sessions. Our approach to respond to frustration has a relatively higher impact on the students whose performance in the sessions is low and the students who spend more time in answering the questions in the Mindspark session.

The results provide an evidence that the approach, detecting and responding to frustration, significantly reduces the students' frustration. The reduction in students' frustration can lead the students to continue learning instead of dropping out the session. Moreover, the motivational messages help the students to create new goals while interacting with the ITS, thus enabling to avoid the negative consequences of frustration, such as dropping out of Mindspark.

8.2 Contributions

In this section, we list the contributions from our research work. The major contributions from our research work are a theory-driven approach to detect affective states, a frustration model to detect frustration with the reasons for that state, an approach to respond to the frustration and an insight into the impact of motivational messages to frustration instances.

- Theory-driven Approach: We developed an approach to detect affective states using data from the students' interaction with the system. Our approach uses only the data from log files, hence, it can be implemented in the large scale deployment of ITS. We have tested our approach on a math ITS to detect frustration. Moreover, we validated the likelihood of generalizing the theory-driven approach to detect other affective states by creating a model to detect boredom in an ITS.

- Frustration Model: We developed a linear regression model to detect frustration in a math ITS – Mindspark, using the theory-driven approach. The theoretical definition of frustration is operationalized for the ITS log data. The detection accuracy of our model is comparatively equal to the existing approaches to detect frustration. Additionally, our model provides the reasons for the frustration of the students. Based on our results, the linear regression model with a ten-fold cross-validation method performs relatively better than other complex classifier models like Support Vector Machine (SVM) and Logistic Model Tree (LMT).

- Respond to Frustration: We provided an approach to avoid the negative consequences of frustration, such as dropping out, by using the motivational messages. The messages to respond to frustration are created based on the reasons for frustration. The software code to detect and respond to frustration is added to the Mindspark's adaptive logic. Students' frustration instances were detected and motivational messages were provided to avoid frustration in real-time. The impact of motivational messages was analyzed and it was found that our approach significantly reduced the number of frustrations per session.

- Impact of Motivational Messages: We provide that the motivational messages to respond to frustration had a relatively high impact on the following kinds of students: (a) those who spent more time in answering the questions; (b) those who had a low average score in Mindspark sessions; (c) those who had a high number of frustration instances per session, before implementing the algorithm.

8.3 Future Work

In this section, we discuss the possible future work from our research. The typical extension of our theory-driven approach is to detect other affective states like confusion and surprise using theoretical definitions and to study the interaction among affective states. Later, we aim to create a comprehensive model to detect all cognitive affective states of the students for adaptation. The model will help in detecting the students' affective state and provide a possibility of adapting the learning content based on specific affective states, thus improving the students' interaction with the ITS/Gaming system, using learning analytics.

In order to create a comprehensive model, the first step will be based on our research outcome–that a theory-driven model can perform equal to data-driven models and also show the reasons for frustration. We plan to apply the theory-driven research method to other cognitive affective states such as boredom, confusion and delight and in turn to identify the features associated with them. We also plan to detect the other affective states using an independent method such as human observation, self reporting and the like. Using the features extracted and the independent data of emotions, we plan to train the theoretical model to identify affective states. The next step will be the forming of the Bayesian network for each of the affective states using the associated features. The interaction among affective states can be modeled as a transition matrix of affective states. Using the transition matrix and features associated with the affective states, one can model the Hidden Markov Model (HMM) for cognitive affective states. In HMM, the affective state at time n is dependent on the affective state at time n-1 and the features *f1, f2, f3* of time n. Hence, the detection accuracy of affective states will be improved. HMM will provide details about the interaction between affective states and the features associated with each affective state. Hence, the adaptation logic can be enhanced for the better performance.

The second possible extension of research work is to investigate whether adapting the learning content based on students' affective states improves the students' learning or not. The first step in this research will be, detecting different cognitive affective states like boredom, confusion, engagement and the like. The next step will be to perform the correlation analysis of each of the affective states with the student learning. If the results suggest that there is a negative correlation between negative affective states like confusion and boredom, then we aim to change the adaptation logic or create strategies to cope with these affective states. Lastly, we plan to analyze whether adapting the learning content or strategies to avoid negative affective states has an impact on student learning or not.

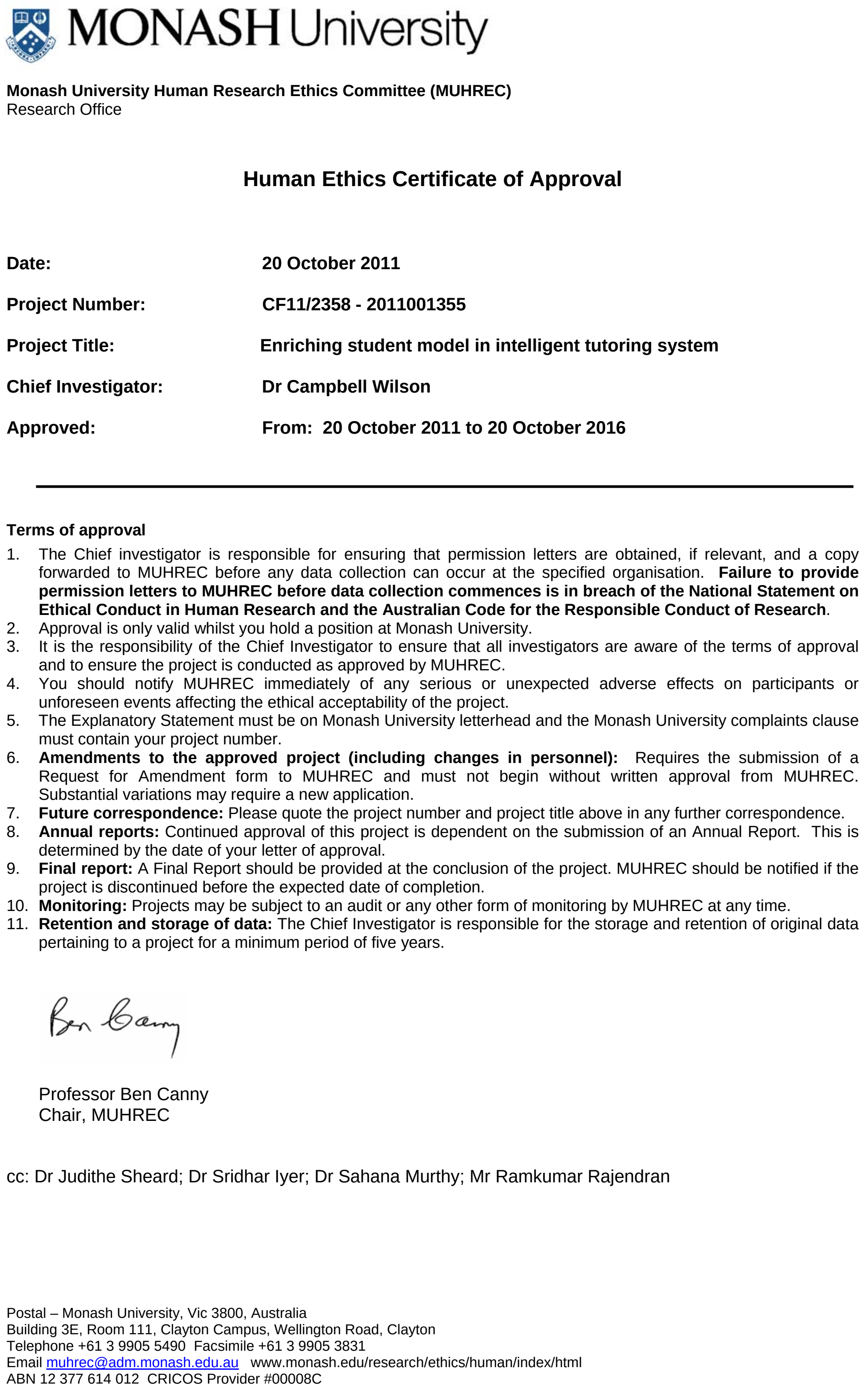
Another possible extension to our research is developing a hybrid system which includes web-cam, eye-movement tracking system, pressure sensitive chairs, etc. which are not expensive and non-intrusive to capture the student's goals and emotions. Such a type of system can improve the performance of the theory-driven approach and would address the limitation of our approach in scalability.

Specific to Mindspark, the course of future work could be the generalization of the frustration model in Mindspark for the different age groups. For this, more data has to be collected from students of different age groups and schools. We also aim to work on detecting different affective states using our theory-driven approach in Mindspark and changing the Mindspark adaptation logic to help students to avoid these affective states.

Appendix

Ethics Documents

Ethics Committee approval, school and company permission for data collection, consent forms for students and parents, and explanation statement for students and general audience are given in this section.



MONASH University

Monash University Human Research Ethics Committee (MUHREC)
Research Office

Human Ethics Certificate of Approval

Date:	**20 October 2011**
Project Number:	**CF11/2358 - 2011001355**
Project Title:	**Enriching student model in intelligent tutoring system**
Chief Investigator:	**Dr Campbell Wilson**
Approved:	**From: 20 October 2011 to 20 October 2016**

Terms of approval

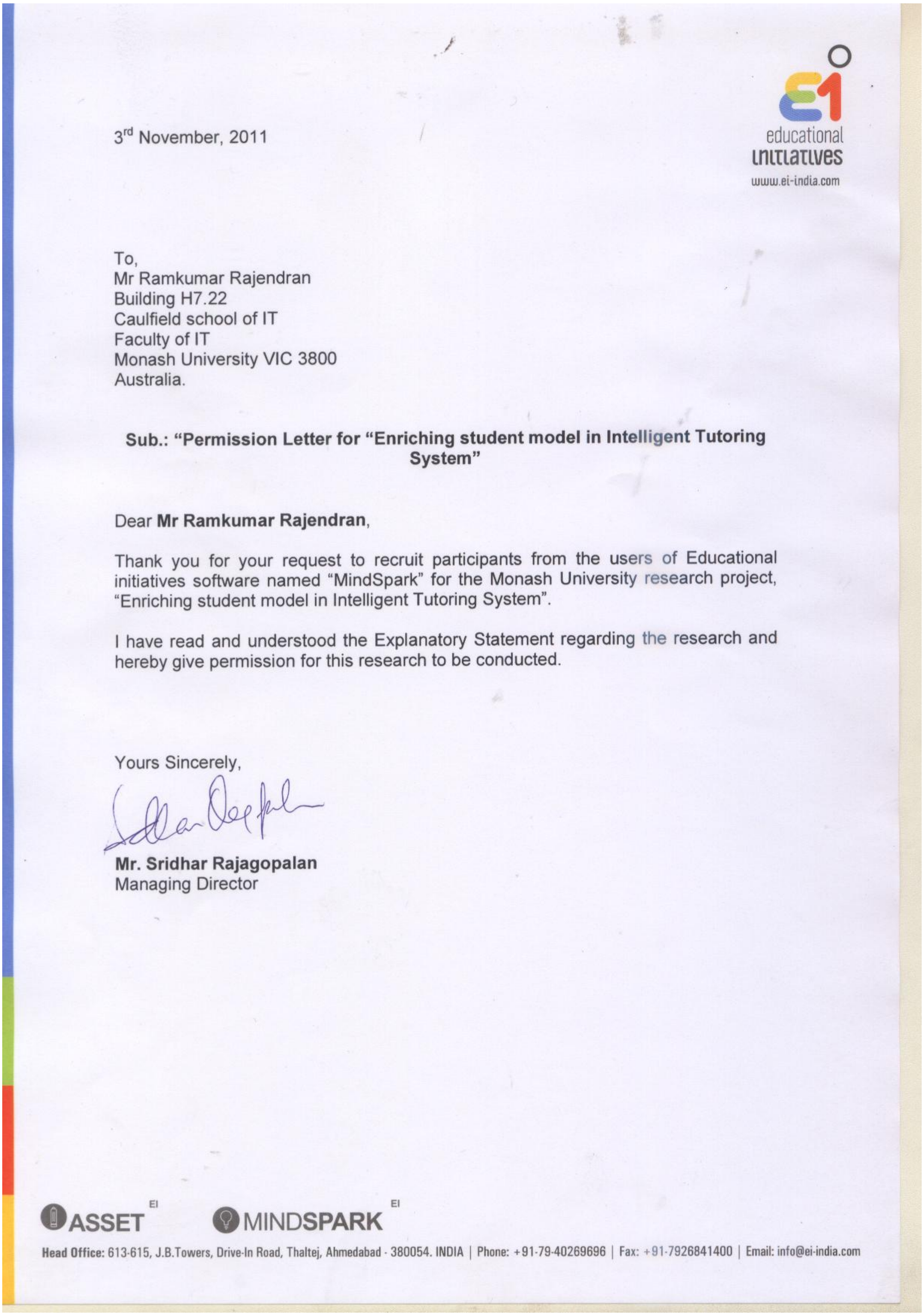
1. The Chief investigator is responsible for ensuring that permission letters are obtained, if relevant, and a copy forwarded to MUHREC before any data collection can occur at the specified organisation. **Failure to provide permission letters to MUHREC before data collection commences is in breach of the National Statement on Ethical Conduct in Human Research and the Australian Code for the Responsible Conduct of Research**.
2. Approval is only valid whilst you hold a position at Monash University.
3. It is the responsibility of the Chief Investigator to ensure that all investigators are aware of the terms of approval and to ensure the project is conducted as approved by MUHREC.
4. You should notify MUHREC immediately of any serious or unexpected adverse effects on participants or unforeseen events affecting the ethical acceptability of the project.
5. The Explanatory Statement must be on Monash University letterhead and the Monash University complaints clause must contain your project number.
6. **Amendments to the approved project (including changes in personnel):** Requires the submission of a Request for Amendment form to MUHREC and must not begin without written approval from MUHREC. Substantial variations may require a new application.
7. **Future correspondence:** Please quote the project number and project title above in any further correspondence.
8. **Annual reports:** Continued approval of this project is dependent on the submission of an Annual Report. This is determined by the date of your letter of approval.
9. **Final report:** A Final Report should be provided at the conclusion of the project. MUHREC should be notified if the project is discontinued before the expected date of completion.
10. **Monitoring:** Projects may be subject to an audit or any other form of monitoring by MUHREC at any time.
11. **Retention and storage of data:** The Chief Investigator is responsible for the storage and retention of original data pertaining to a project for a minimum period of five years.

Ben Canny

Professor Ben Canny
Chair, MUHREC

cc: Dr Judithe Sheard; Dr Sridhar Iyer; Dr Sahana Murthy; Mr Ramkumar Rajendran

Postal – Monash University, Vic 3800, Australia
Building 3E, Room 111, Clayton Campus, Wellington Road, Clayton
Telephone +61 3 9905 5490 Facsimile +61 3 9905 3831
Email muhrec@adm.monash.edu.au www.monash.edu/research/ethics/human/index/html
ABN 12 377 614 012 CRICOS Provider #00008C



educational initiatives
www.ei-india.com

3rd November, 2011

To,
Mr Ramkumar Rajendran
Building H7.22
Caulfield school of IT
Faculty of IT
Monash University VIC 3800
Australia.

Sub.: "Permission Letter for "Enriching student model in Intelligent Tutoring System"

Dear **Mr Ramkumar Rajendran**,

Thank you for your request to recruit participants from the users of Educational initiatives software named "MindSpark" for the Monash University research project, "Enriching student model in Intelligent Tutoring System".

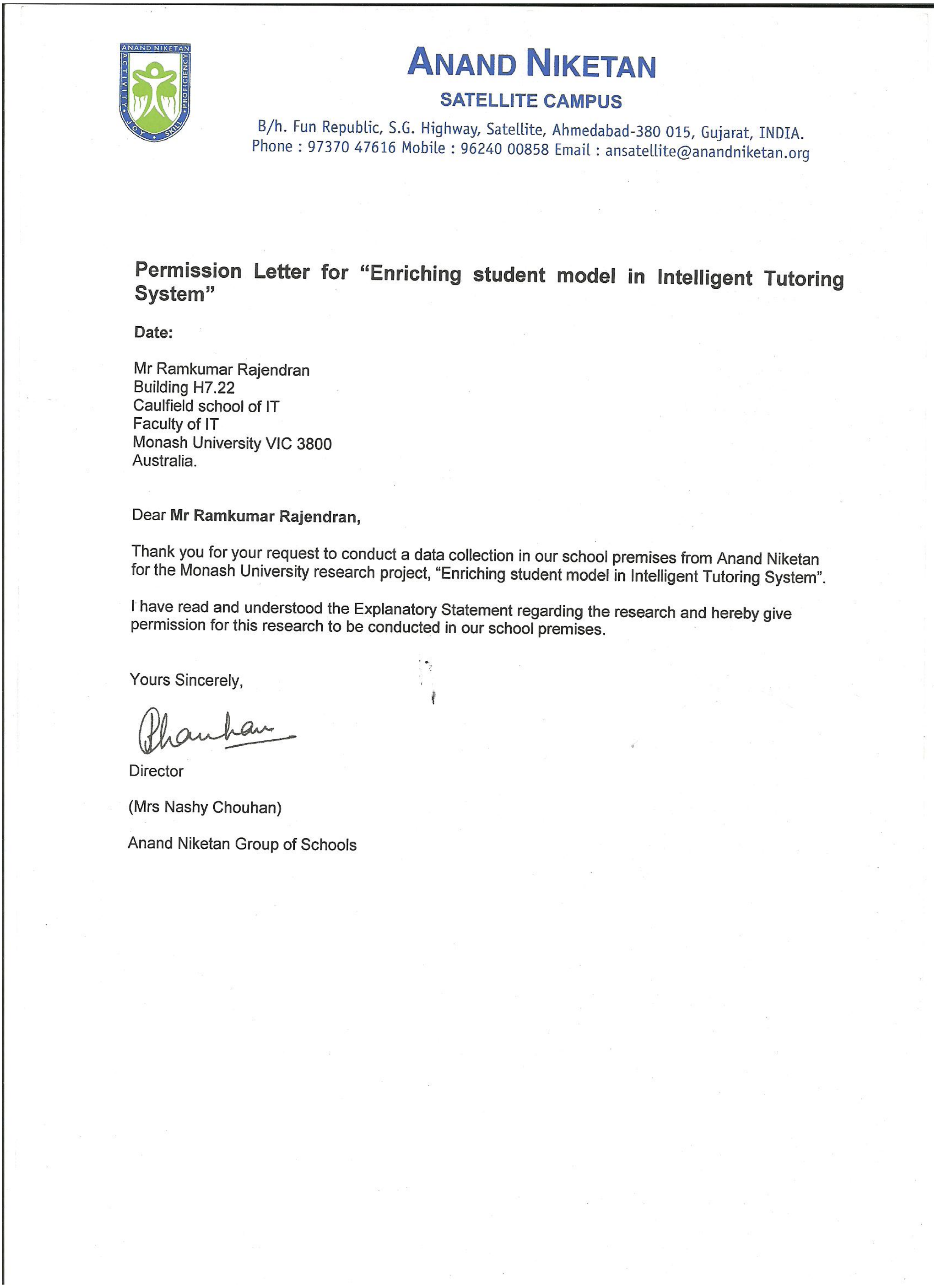
I have read and understood the Explanatory Statement regarding the research and hereby give permission for this research to be conducted.

Yours Sincerely,

Mr. Sridhar Rajagopalan
Managing Director

ASSET EI MINDSPARK EI

Head Office: 613-615, J.B.Towers, Drive-In Road, Thaltej, Ahmedabad - 380054. INDIA | Phone: +91-79-40269696 | Fax: +91-7926841400 | Email: info@ei-india.com



ANAND NIKETAN

SATELLITE CAMPUS

B/h. Fun Republic, S.G. Highway, Satellite, Ahmedabad-380 015, Gujarat, INDIA.
Phone : 97370 47616 Mobile : 96240 00858 Email : ansatellite@anandniketan.org

Permission Letter for "Enriching student model in Intelligent Tutoring System"

Date:

Mr Ramkumar Rajendran
Building H7.22
Caulfield school of IT
Faculty of IT
Monash University VIC 3800
Australia.

Dear **Mr Ramkumar Rajendran,**

Thank you for your request to conduct a data collection in our school premises from Anand Niketan for the Monash University research project, "Enriching student model in Intelligent Tutoring System".

I have read and understood the Explanatory Statement regarding the research and hereby give permission for this research to be conducted in our school premises.

Yours Sincerely,

Chauhan

Director

(Mrs Nashy Chouhan)

Anand Niketan Group of Schools

Consent Form - Students

Title: Enriching student model in Intelligent Tutoring System

NOTE: This consent form will remain with the Monash University researcher for their records

I agree to take part in the Monash University research project, “Enriching student model in Intelligent Tutoring System”. I have had the project explained to me, and I have read the Explanatory Statement, which I will keep for my records. I understand that agreeing to take part means that:

I agree to allow the researcher to video-tape my facial expressions while interacting with Mindspark, , a math system which I use in my math class. □ Yes □ No

I agree to record my use of Mindspark in a computer log file □ Yes □ No

and

I understand that my participation is voluntary, that I can choose not to participate in the project, and that I can stop at any stage of the project without being penalised or disadvantaged in any way.

and

I understand that any data that the researcher extracts from the video-tape/ computer log files for use in reports or published findings will not, under any circumstances, contain my name or identifying characteristics.

and

I understand that data from the video–tape and computer log files will be kept in a secure storage and accessible only to the research team. I also understand that the data will be destroyed after a 5 year period.

Participant’s name

Signature

Date

Consent Form - Parents

Title: Enriching student model in Intelligent Tutoring System

NOTE: Signed written consent will remain with the Monash University researcher for their records.

I agree that _________________________ (insert full name of participant) may take part in the Monash University research project, "Enriching student model in Intelligent Tutoring System". The project has been explained to _______________ (insert name of participant) and to me, and I have read the Explanatory Statement, which I will keep for my records.

I understand that agreeing to take part means that I am willing to allow __________ (insert full name of participant) to:

- Video-tape his/her facial expressions while interacting with Mindspark, a math Intelligent Tutoring System used in his/her regular math class □ Yes □ No
- Record the interactions with Mindspark, in a computer log file □ Yes □ No

Participant's name:

Participant's Age:

Parent's / Guardian's Name:

Parent's / Guardian's relationship to participant:

Parent's / Guardian's Signature:

Date

MONASH University

Date

Explanatory Statement - parents

Title: **Enriching student model in Intelligent Tutoring System**

This information sheet is for you to keep.

My name is Ramkumar Rajendran and I am conducting a research project with Dr Campbell Wilson, a senior lecturer in the Faculty of Information technology, Monash University, and Dr. Sridhar Iyer, Assoc Prof in computer science and engineering department of IIT Bombay, towards a PhD at IITB-Monash Research Academy, a joint program of Indian Institute of Technology Bombay, India and Monash University, Australia. This means that I will be writing a thesis which is the equivalent of a 250 page book.

Please read this Explanatory Statement in full before making a decision.

I am looking for the students who use Mindspark, a math Intelligent Tutoring System (ITS), in their school as part of their curriculum. We obtained the information about the users of Mindspark from the company “Educational Initiatives (EI)”, India.

The aim of the research

The aim of this research is to identify a set of affective states (namely confusion, frustration and boredom) from data collected on computer log files. The log files contain data from the students’ interaction with ITS.

The affective states identified from the log file data will be validated through comparison with the results obtained from observation of students’ facial expressions while they interact with ITS.

Possible benefits

Participants will not directly benefit from this research. If our model identifies the affective states automatically only using the student interaction with ITS, it will help to improve the function of ITS thereby student learning experience.

Participation Involves:

The study involves video recording the participants’ facial expressions while they interact with ITS.
The participants’ interaction with ITS is recorded in computer log files.

Time for this research:

Participants do not need to spend extra time for this research; participants are video recorded during their one regular math tutoring session which will be 40 minutes.

Participation and Withdrawal:

Participation in this study is completely voluntary and the participant is free to choose whether to be or not part of this study. Parents can decide whether their child participates in this study or not. If parents consent to their child participating, they may withdraw their child from the study at any stage. Furthermore, if parents give consent to their child participating, then their child can withdraw from the study at any stage.

Confidentiality

We obtain student information as de-identified data from database, so personal information will not be stored in the data.

Storage of data

Data collected will be stored in accordance with Monash University regulations, kept on University premises, in a locked room for 5 years. A report of the study may be submitted for publication in scientific conferences, but individual participants will not be identifiable in such a report.

Results

If you would like to be informed of the aggregate research finding, please contact Ramkumar Rajendran on +61 3 990 31217 or email **rraj28@student.monash.edu**.

If you would like to contact the researchers about any aspect of this study, please contact the Chief Investigator:	If you have a complaint concerning the manner in which this research **CF11/2358 - 2011001355** is being conducted, please contact:
Dr Campbell Wilson Deputy Head of School, Faculty of Information Technology, Caulfield Campus, Monash University PO Box 197, Caulfield East Victoria 3145, Australia. Email: **Campbell.Wilson@monash.edu** Phone: +61 3 990 31142	**Executive Officer** **Monash University Human Research Ethics Committee (MUHREC)** **Building 3e Room 111** **Research Office** **Monash University VIC 3800** **Tel: +61 3 9905 2052 Fax: +61 3 9905 3831** **Email: muhrec@monash.edu**

Thank you.

Ramkumar Rajendran

MONASH University

Date

Explanatory Statement - Students

Title: **Enriching student model in Intelligent Tutoring System**

This information sheet is for you to keep.

1. My name is Ramkumar Rajendran and I am conducting a research project with Dr Campbell Wilson, a senior lecturer at Monash University, Australia and Associate Professor Sridhar Iyer, IIT Bombay, India. This research is for my PhD studies at IITB-Monash Research Academy.
2. Please read this Explanatory Statement in full before making a decision.
3. We are asking you take part in a research study because we are trying to automatically identify emotions while you use Mindspark in your math class.
4. If you agree to be in this study,
   a. We will record your facial expressions while you use Mindspark.
   b. Your use of Mindspark will be recorded on the computer.
5. We will conduct this study in your regular math class. It will not take any extra time to participate in this study.
6. Please talk to your parents before you decide whether or not to participate. We also explained this research study to your parents and asked them to give permission for you to take part of this study. But even if your parents agree, you can still decide not to do this.
7. If you don't want to be in this study, you don't have to participate. There is no compulsion, and no one will upset if you do not participate. However, if you do agree to participate, you may stop participation at any time during the session.
8. If you have questions, you can ask your math teacher. If you have any question later that you did not think of now, you can call me on +61 3 9903 1217 or email me at **rraj28@student.monash.edu**.

If you would like to contact the researchers about any aspect of this study, please contact the Chief Investigator:	If you have a complaint concerning the manner in which this research **CF11/2358 - 2011001355** is being conducted, please contact:
Dr Campbell Wilson Deputy Head of School, Faculty of Information Technology, Caulfield Campus, Monash University PO Box 197, Caulfield East Victoria 3145, Australia. Email: **Campbell.Wilson@monash.edu** Phone: +61 3 990 31142	**Executive Officer** **Monash University Human Research Ethics Committee (MUHREC)** **Building 3e Room 111** **Research Office** **Monash University VIC 3800** **Tel: +61 3 9905 2052 Fax: +61 3 9905 3831** **Email: muhrec@monash.edu**

Thank you.

Ramkumar Rajendran

Bibliography

[1] Mindspark: Improve your math skills. `http://www.mindspark.in/`. Accessed Feb 20, 2014.

[2] M. Adibuzzaman, N. Jain, N. Steinhafel, M. Haque, F. Ahmed, S. I. Ahamed, and R. Love. Towards in situ affect detection in mobile devices: a multimodal approach. In *Proceedings of the 2013 Research in Adaptive and Convergent Systems*, pages 454–460. ACM, 2013.

[3] L. Agnieszka. Affect-awareness framework for intelligent tutoring systems. In *Human System Interaction (HSI), 2013 The 6th International Conference on*, pages 540–547. IEEE, 2013.

[4] C. O. Alm, D. Roth, and R. Sproat. Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 579–586. Association for Computational Linguistics, 2005.

[5] A. Amsel. Frustration theory: Many years later. *Psychological bulletin*, 112(3):396–399, 1992.

[6] J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier. Cognitive tutors: Lessons learned. *The journal of the learning sciences*, 4(2):167–207, 1995.

[7] H. Anger Elfenbein and N. Ambady. On the universality and cultural specificity of emotion recognition : A meta-analysis. *Psychology Bulletin*, 128(2):203–235, 2002.

[8] I. Arroyo, D. G. Cooper, W. Burleson, B. P. Woolf, K. Muldner, and R. Christopherson. Emotion sensors go to school. In *Artificial Intelligence in Education*, volume 200, pages 17–24, 2009.

[9] I. Arroyo and B. P. Woolf. Inferring learning and attitudes from a bayesian network of log file data. In *AIED*, pages 33–40, 2005.

[10] J. W. Atkinson. *An introduction to motivation.* Van Nostrand, 1964.

[11] R. S. J. d. Baker, S. M. Gowda, M. Wixon, J. Kalka, A. Z. Wagner, A. Salvi, V. Aleven, G. W. Kusbit, J. Ocumpaugh, and L. Rossi. Towards sensor-free affect detection in cognitive tutor algebra. *International Educational Data Mining Society*, 2012.

[12] T. Bänziger, D. Grandjean, and K. R. Scherer. Emotion recognition from expressions in face, voice,

and body: the multimodal emotion recognition test (mert). *Emotion*, 9(5):691–704, 2009.

[13] J. E. Barmack. The effect of benzedrine sulfate (benzyl methyl carbinamine) upon the report of boredom and other factors. *The Journal of Psychology*, 5(1):125–133, 1938.

[14] M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Fully automatic facial action recognition in spontaneous behavior. In *7th International Conference on Automatic Face and Gesture Recognition.*, pages 223–230, 2006.

[15] S. Batool, M. I. Yousuf, and Q. Parveen. A study of attribution patterns among high and low attribution groups: An application of weiners attribution theory. *Anthropologist*, 14(3):193–197, 2012.

[16] C. Beal, J. Beck, and B. Woolf. Impact of intelligent computer instruction on girls math self concept and beliefs in the value of math. In *Poster presented at the annual meeting of the American Educational Research Association, San Diego*, 1998.

[17] N. Bianchi-Berthouze and C. L. Lisetti. Modeling multimodal expression of users affective subjective experience. *User Modeling and User-Adapted Interaction*, 12(1):49–84, 2002.

[18] K. Brawner and B. Goldberg. Real-time monitoring of ecg and gsr signals during computer-based training. In *Intelligent Tutoring Systems*, pages 72–77, 2012.

[19] E. J. Brown, T. J. Brailsford, T. Fisher, and A. Moore. Evaluating learning style personalization in adaptive systems: Quantitative methods and approaches. *IEEE Transactions on Learning Technologies*, 2:10–22, 2009.

[20] J. S. Brown, R. R. Burton, and A. G. Bell. Sophie: A step toward creating a reactive learning environment. *International Journal of Man-Machine Studies*, 7(5):675–696, 1975.

[21] P. Brusilovsky. Adaptive hypermedia. *User modeling and user-adapted interaction*, 11(1-2):87–110, 2001.

[22] P. Brusilovsky and E. Milln. User models for adaptive hypermedia and adaptive educational systems. In *The Adaptive Web, LNCS 4321*, pages 3–53, 2007.

[23] P. Brusilovsky, E. Schwarz, and G. Weber. Elm-art: An intelligent tutoring system on world wide web. In *Intelligent tutoring systems*, pages 261–269, 1996.

[24] R. R. Burton and J. S. Brown. An investigation of computer coaching for informal learning activities. *International Journal of Man-Machine Studies*, 11(1):5–24, 1979.

[25] R. A. Calvo, I. Brown, and S. Scheding. Effect of experimental factors on the recognition of affective

mental states through physiological measures. In *AI 2009: Advances in Artificial Intelligence*, pages 62–70. Springer, 2009.

[26] R. A. Calvo and S. K. D'Mello. Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on Affective Computing*, 1:18–37, 2010.

[27] N. Capuano, M. Marsella, and S. Salerno. Abits: An agent based intelligent tutoring system for distance learning. In *Proceedings of the International Workshop on Adaptive and Intelligent Web-Based Education Systems, ITS*, 2000.

[28] J. S. Carriere, J. A. Cheyne, and D. Smilek. Everyday attention lapses and memory failures: The affective consequences of mindlessness. *Consciousness and cognition*, 17(3):835–847, 2008.

[29] R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168. ACM, 2006.

[30] L.-H. Chen, Y.-C. Lai, and Y.-H. Weng. Intelligent e-learning system with personalized misconception diagnose and learning path guidance. In *International Conference on Electronic Business*, 2009.

[31] W. J. Clancey. Tutoring rules for guiding a case method dialogue. *International Journal of Man-Machine Studies*, 11(1):25–49, 1979.

[32] W. J. Clancey. Intelligent tutoring systems: A tutorial survey. Technical report, DTIC Document, 1986.

[33] C. Conati and H. Maclaren. Empirically building and evaluating a probabilistic model of user affect. *User Modeling and User-Adapted Interaction*, 19(3):267–303, 2009.

[34] D. Cooper, I. Arroyo, B. Woolf, K. Muldner, W. Burleson, and R. Christopherson. Sensors model student self concept in the classroom. In *User Modeling, Adaptation, and Personalization*, pages 30–41. Springer, 2009.

[35] A. T. Corbett, K. R. Koedinger, and J. R. Anderson. Intelligent tutoring systems. *Handbook of humancomputer interaction*, pages 849–874, 1997.

[36] M. Coulson. Attributing emotion to static body postures: Recognition accuracy, confusions, and viewpoint dependence. *Journal of nonverbal behavior*, 28(2):117–139, 2004.

[37] M. Csikszentmihalyi. *Finding flow: The psychology of engagement with everyday life*. Basic Books, 1997.

[38] R. S. J. d Baker, A. T. Corbett, and V. Aleven. More accurate student modeling through contextual

estimation of slip and guess probabilities in bayesian knowledge tracing. In *Intelligent Tutoring Systems*, pages 406–415. Springer, 2008.

[39] R. S. J. d. Baker, S. K. D'Mello, M. M. T. Rodrigo, and A. C. Graesser. Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive-affective states during interactions with three different computer-based learning environments. *International Journal of Human-Computer Studies*, 68(4):223 – 241, 2010.

[40] T. Danisman and A. Alpkocak. Feeler: Emotion classification of text using vector space model. In *AISB 2008 Convention Communication, Interaction and Social Intelligence*, volume 2, pages 53–59, 2008.

[41] C. Darwin. *The expression of the emotions in man and animals.* Oxford University Press, 1998.

[42] E. L. Deci and R. M. Ryan. *Self-Determination.* Wiley Online Library, 2010.

[43] C. Dede, M. C. Salzman, and R. B. Loftin. Sciencespace: Virtual realities for learning complex and abstract scientific concepts. In *Virtual Reality Annual International Symposium, 1996., Proceedings of the IEEE 1996*, pages 246–252, 1996.

[44] S. DMello, R. Picard, and A. Graesser. Towards an affect-sensitive autotutor. *IEEE Intelligent Systems*, 22(4):53–61, 2007.

[45] S. K. D'Mello, S. D. Craig, K. Fike, and A. C. Graesser. Responding to learners' cognitive-affective states with supportive and shakeup dialogues. In *Human-Computer Interaction. Ambient, Ubiquitous and Intelligent Interaction*, pages 595–604, 2009.

[46] S. K. D'Mello, S. D. Craig, B. Gholson, and S. Franklin. Integrating affect sensors in an intelligent tutoring system. In *In Affective Interactions: The Computer in the Affective Loop Workshop at 2005 Intl. Conf. on Intelligent User Interfaces, 2005*, pages 7–13. AMC Press, 2005.

[47] S. K. D'Mello, S. D. Craig, J. Sullins, and A. C. Graesser. Predicting affective states expressed through an emote-aloud procedure from autotutor's mixed-initiative dialogue. *Int. J. Artif. Intell. Ed.*, 16:3–28, January 2006.

[48] S. K. D'Mello, S. D. Craig, A. Witherspoon, B. Mcdaniel, and A. Graesser. Automatic detection of learner's affect from conversational cues. *User Modeling and User-Adapted Interaction*, 18:45–80, February 2008.

[49] S. K. D'Mello and A. C. Graesser. Automatic detection of learner's affect from gross body language. *Applied Artificial Intelligence*, 23(2), 2009.

[50] J. Dollard, N. E. Miller, L. W. Doob, O. H. Mowrer, and R. R. Sears. Frustration and aggression. 1939.

[51] G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski. Classifying facial actions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(10):974–989, 1999.

[52] T. Dragon, I. Arroyo, B. P. Woolf, W. Burleson, R. Kaliouby, and H. Eydgahi. Viewing student affect and learning through classroom observation and physical sensors. In *Proceedings of the 9th international conference on Intelligent Tutoring Systems*, ITS '08, pages 29–39, 2008.

[53] C. S. Dweck. Motivational processes affecting learning. *American psychologist*, 41(10):1040–1048, 1986.

[54] C. S. Dweck. Messages that motivate: How praise molds students' beliefs, motivation, and performance (in surprising ways). In *Improving Academic Achievement*, pages 37 – 60. Academic Press, 2002.

[55] P. Ekman. Universals and cultural differences in facial expressions of emotion. In *Nebraska symposium on motivation*. University of Nebraska Press, 1971.

[56] P. Ekman. Strong evidence for universals in facial expressions: a reply to Russell's mistaken critique. *Psychology Bulletin*, 115(2):268–287, 1994.

[57] P. Ekman and W. V. Friesen. *Unmasking the face: A guide to recognizing emotions from facial clues*. Malor Books, 2003.

[58] R. El Kaliouby and P. Robinson. *Real-time inference of complex mental states from facial expressions and head gestures*. Springer, 2005.

[59] C. Eliot and B. P. Woolf. Reasoning about the user within a simulation-based real-time training system. In *Fourth International Conference on User Modeling, Hyannis Cape Cod, Mass*, pages 15–19, 1994.

[60] P. C. Ellsworth and K. R. Scherer. Appraisal processes in emotion. *Handbook of affective sciences*, pages 572–595, 2003.

[61] H. A. Feild, J. Allan, and R. Jones. Predicting searcher frustration. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 34–41. ACM, 2010.

[62] F. Försterling. Attributional retraining: A review. *Psychological Bulletin*, 98(3):495, 1985.

[63] M. Gagné and E. L. Deci. Self-determination theory and work motivation. *Journal of Organizational*

behavior, 26(4):331–362, 2005.

[64] J. P. Gee. What video games have to teach us about learning and literacy. *Computers in Entertainment (CIE)*, 1(1):20–23, 2003.

[65] J. P. Gee. Learning and games. *The ecology of games: Connecting youth, games, and learning*, 3:21–40, 2008.

[66] S. Geisser. *Predictive Inference: An Introduction.* Monographs on Statistics and Applied Probability. Chapman & Hall, 1993.

[67] T. Gordon. *PET: Parent Effectiveness Training.* New American Library, 1970.

[68] A. C. Graesser, P. Chipman, B. C. Haynes, and A. Olney. Autotutor: An intelligent tutoring system with mixed-initiative dialogue. *Education, IEEE Transactions on*, 48(4):612–618, 2005.

[69] A. C. Graesser, K. VanLehn, C. P. Rosé, P. W. Jordan, and D. Harter. Intelligent tutoring systems with conversational dialogue. *AI magazine*, 22(4):39–51, 2001.

[70] A. C. Graesser, K. Wiemer-Hastings, P. Wiemer-Hastings, and R. Kreuz. Autotutor: A simulation of a human tutor. *Cognitive Systems Research*, 1(1):35–51, 1999.

[71] J. F. Grafsgaard, J. B. Wiggins, K. E. Boyer, E. N. Wiebe, and J. C. Lester. Automatically recognizing facial expression: Predicting engagement and frustration. In *Proceedings of the 6th International Conference on Educational Data Mining*, 2013.

[72] S. Graham. A review of attribution theory in achievement contexts. *Educational Psychology Review*, 3(1):5–39, 1991.

[73] S. Graham, B. Weiner, D. Berliner, and R. Calfee. Theories and principles of motivation. *Handbook of educational psychology*, 4:63–84, 1996.

[74] H. Gunes and M. Piccardi. Bi-modal emotion recognition from expressive face and body gestures. *Journal of Network and Computer Applications*, 30(4):1334–1345, 2007.

[75] D. F. Halpern, J. Aronson, N. Reimer, S. Simpkins, J. R. Star, and K. Wentzel. Encouraging girls in math and science. 2007.

[76] J. Hartley and D. H. Sleeman. Towards more intelligent teaching systems. *International Journal of Man-Machine Studies*, 5(2):215–236, 1973.

[77] F. Heider. *The Psychology of Interpersonal Relations.* Lawrence Erlbaum Associates, 1958.

[78] J. D. Hollan, E. L. Hutchins, and L. Weitzman. Steamer: An interactive inspectable simulation-

based training system. *AI magazine*, 5(2):15–27, 1984.

[79] K. Hone. Empathic agents to reduce user frustration: The effects of varying agent characteristics. *Interacting with Computers*, 18(2):227–245, 2006.

[80] C.-M. Hong, C.-M. Chen, M.-H. Chang, and S.-C. Chen. Intelligent web-based tutoring system with personalized learning path guidance. *Advanced Learning Technologies, IEEE International Conference on*, 0:512–516, 2007.

[81] C. Hull. Principles of behavior. 1943.

[82] M. S. Hussain, O. AlZoubi, R. A. Calvo, and S. K. D'Mello. Affect detection from multichannel physiology during learning sessions with autotutor. In *Proceedings of the 15th international conference on Artificial intelligence in education*, AIED'11, pages 131–138, 2011.

[83] M. S. Hussain, H. Monkaresi, and R. A. Calvo. Categorical vs. dimensional representations in multimodal affect detection during learning. In *Intelligent Tutoring Systems*, pages 78–83, 2012.

[84] C. E. Izard. *The face of emotion.* Appleton-Century-Crofts, 1971.

[85] C. E. Izard. Innate and universal facial expressions: evidence from developmental and cross-cultural research. *Psychological Bulletin*, 115(2):288–299, 1994.

[86] M. C. Jadud. *An exploration of novice compilation behaviour in BlueJ.* PhD thesis, University of Kent, 2006.

[87] T. Kanade, Y. Tian, and J. F. Cohn. Comprehensive database for facial expression analysis. In *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 46–53, 2000.

[88] A. Kapoor, W. Burleson, and R. W. Picard. Automatic prediction of frustration. *Int. J. Hum.-Comput. Stud.*, 65:724–736, August 2007.

[89] A. Kapoor and R. W. Picard. Multimodal affect recognition in learning environments. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 677–682, 2005.

[90] S. Katz and A. Lesgold. The role of the tutor in computer-based collaborative learning situations. *Computers as cognitive tools*, pages 289–317, 1993.

[91] F. A. Khan, S. Graf, E. R. Weippl, and A. M. Tjoa. An approach for identifying affective states through behavioral patterns in web-based learning management systems. In *Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services*, iiWAS '09, pages 431–435, 2009.

[92] K. H. Kim, S. Bang, and S. Kim. Emotion recognition system using short-term monitoring of physiological signals. *Medical and biological engineering and computing*, 42(3):419–427, 2004.

[93] J. Klein, Y. Moon, and R. W. Picard. This computer responds to user frustration: Theory, design, and results. *Interacting with Computers*, 14:119–140, Jan 2002.

[94] K. R. Koedinger, J. R. Anderson, W. H. Hadley, M. A. Mark, et al. Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education (IJAIED)*, 8:30–43, 1997.

[95] B. Kort, R. Reilly, and R. W. Picard. An affective model of interplay between emotions and learning: Reengineering educational pedagogy-building a learning companion. In *Advanced Learning Technologies, IEEE International Conference on*, pages 0043–0046. IEEE Computer Society, 2001.

[96] S. P. Lajoie and A. Lesgold. Apprenticeship training in the workplace: Computer-coached practice environment as a new form of apprenticeship. *Machine-Mediated Learning*, 3(1):7–28, 1989.

[97] R. Lawson. *Frustration; the development of a scientific concept.* Critical issues in psychology series. Macmillan, 1965.

[98] J. Lazar, A. Jones, M. Hackley, and B. Shneiderman. Severity and impact of computer user frustration: A comparison of student and workplace users. *Interact. Comput.*, 18(2):187–207, mar 2006.

[99] R. S. Lazarus. *Emotion and adaptation.* Oxford University Press New York, 1991.

[100] C. M. Lee and S. S. Narayanan. Toward detecting emotions in spoken dialogs. *Speech and Audio Processing, IEEE Transactions on*, 13(2):293–303, 2005.

[101] B. Lehman, M. Matthews, S. DMello, and N. Person. What are you feeling? investigating student affective states during expert human tutoring sessions. In *Intelligent Tutoring Systems*, pages 50–59. Springer, 2008.

[102] C. Leung, E. Wai, and Q. Li. An experimental study of a personalized learning environment through open-source software tools. *IEEE Transactions on Education*, Vol 50(No. 4), Nov 2007.

[103] W. Li and H. Xu. Text-based emotion classification using emotion cause extraction. *Expert Systems with Applications*, (0):–, 2013.

[104] D. J. Litman and K. Forbes-Riley. Predicting student emotions in computer-human tutoring dialogues. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 351–358, 2004.

[105] C. Liu, K. Conn, N. Sarkar, and W. Stone. Physiology-based affect recognition for computer-assisted intervention of children with autism spectrum disorder. *International journal of human-computer studies*, 66(9):662–677, 2008.

[106] E. A. Locke and G. P. Latham. What should we do about motivation theory? six recommendations for the twenty-first century. *Academy of Management Review*, 29(3):388–403, 2004.

[107] M. K. Mandal, M. P. Bryden, and M. B. Bulman-Fleming. Similarities and variations in facial expressions of emotions: Cross-cultural evidence. *International Journal of Psychology*, 31(1):49–58, 1996.

[108] H. Mark, F. Eibe, H. Geoffrey, P. Bernhard, R. Peter, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11, 2009.

[109] S. Marsella, J. Gratch, and P. Petta. Computational models of emotion. *A Blueprint for Affective Computing-A Sourcebook and Manual*, pages 21–46, 2010.

[110] B. T. McDaniel, S. K. D'Mello, B. G. King, P. Chipman, K. Tapp, and A. C. Graesser. Facial features for affective state detection in learning environments. In *Proceedings of the 29th Annual Cognitive Science Society*, pages 467–472, 2007.

[111] S. W. Mcquiggan, S. Lee, and J. C. Lester. Early prediction of student frustration. In *Proceedings of the 2nd international conference on Affective Computing and Intelligent Interaction*, ACII '07, pages 698–709, 2007.

[112] E. Melis and J. Siekmann. Activemath: An intelligent tutoring system for mathematics. In *Artificial Intelligence and Soft Computing-ICAISC 2004*, pages 91–101. Springer, 2004.

[113] D. Michie, D. Spiegelhalter, C. Taylor, and J. Campbell. *Machine learning, neural and statistical classification.* Ellis Horwood London, 1994.

[114] W. L. Mikulas and S. J. Vodanovich. The essence of boredom. *Psychological aspects*, 43(1):3, 1993.

[115] C. T. Morgan, R. A. King, J. R. Weisz, and J. Schopler. *Introduction to Psychology.* McGraw-Hill Book Company, seventh edition edition, 1986.

[116] S. Mota and R. W. Picard. Automated posture analysis for detecting learner's interest level. In *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03. Conference on*, volume 5, pages 49–49. IEEE, 2003.

[117] T. Murray. Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education*, 10:98–129, 1999.

[118] W. R. Nugent and H. Halvorson. Testing the effects of active listening. *Research on Social Work Practice*, 5(2):152–175, 1995.

[119] J. F. O'Hanlon. Boredom: Practical consequences and a theory. *Acta psychologica*, 49(1):53–82, 1981.

[120] A. Ortony, G. L. Clore, and A. Collins. *The cognitive structure of emotions.* Cambridge university press, 1990.

[121] A. Panning, I. Siegert, A. Al-Hamadi, A. Wendemuth, D. Rosner, J. Frommer, G. Krell, and B. Michaelis. Multimodal affect recognition in spontaneous hci environment. In *Signal Processing, Communication and Computing (ICSPCC), 2012 IEEE International Conference on*, pages 430–435, 2012.

[122] M. Pantic and I. Patras. Dynamics of facial expression: recognition of facial actions and their temporal segments from face profile image sequences. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 36(2):433–449, 2006.

[123] T. Partala and V. Surakka. The effects of affective interventions in human–computer interaction. *Interacting with computers*, 16(2):295–309, 2004.

[124] R. Pekrun. The impact of emotions on learning and achievement: Towards a theory of cognitive/motivational mediators. *Applied Psychology*, 41(4):359–376, 1992.

[125] R. W. Picard. Affective computing. Technical report, Cambridge, MA, 1997.

[126] R. W. Picard. *Affective computing.* MIT press, 2000.

[127] C. C. Pinder. *Work motivation in organizational behavior .* Psychology Press, 2008.

[128] E. Popescu. Evaluating the impact of adaptation to learning styles in a web-based educational system. In *ICWL '009: Proceedings of the 8th International Conference on Advances in Web Based Learning*, pages 343–352, 2009.

[129] H. Prendinger and M. Ishizuka. The empathic companion: A character-based interface that addresses users' affective states. *Applied Artificial Intelligence*, 19(3-4):267–285, 2005.

[130] S. Ritter, J. Anderson, M. Cytrynowicz, and O. Medvedeva. Authoring content in the pat algebra tutor. *Journal of Interactive Media in Education*, 98(9), 1998.

[131] S. Ritter, J. R. Anderson, K. R. Koedinger, and A. Corbett. Cognitive tutor: Applied research in mathematics education. *Psychonomic bulletin & review*, 14(2):249–255, 2007.

[132] M. M. T. Rodrigo and R. S. J. d. Baker. Coarse-grained detection of student frustration in an

introductory programming course. In *Proceedings of the fifth international workshop on Computing education research workshop*, ICER '09, pages 75–80, 2009.

[133] M. M. T. Rodrigo, R. S. J. d. Baker, J. Agapito, J. Nabos, M. C. Repalam, S. S. Reyes, and M. O. C. Z. S. Pedro. The effects of an interactive software agent on student affective dynamics while using an intelligent tutoring system. *IEEE Transactions on Affective Computing*, 3(2):224 –236, 2012.

[134] M. M. T. Rodrigo, R. S. J. d. Baker, M. C. V. Lagud, S. A. L. Lim, A. F. Macapanpan, S. A. M. S. Pascua, J. Q. Santillano, L. R. S. Sevilla, J. O. Sugay, S. Tep, and N. J. B. Viehland. Affect and usage choices in simulation problem-solving environments. In *Proceeding of the 2007 conference on Artificial Intelligence in Education: Building Technology Rich Learning Contexts That Work*, pages 145–152, 2007.

[135] I. J. Roseman. Cognitive determinants of emotion: A structural theory. *Review of Personality & Social Psychology*, 1984.

[136] I. J. Roseman, M. S. Spindel, and P. E. Jose. Appraisals of emotion-eliciting events: Testing a theory of discrete emotions. *Journal of Personality and Social Psychology*, 59(5):899–915, 1990.

[137] S. Rosenzweig. A general outline of frustration. *Journal of Personality*, 7(2):151–160, 1938.

[138] S. Rosenzweig and G. Mason. An experimental study of memory in relation to the theory of repression. *British Journal of Psychology. General Section*, 24(3):247–265, 1934.

[139] J. Sabourin, B. Mott, and J. C. Lester. Modeling learner affect with theoretically grounded dynamic bayesian networks. In *International Conference on Affective Computing and Intelligent Interaction*, pages 286–295, 2011.

[140] G. Salton. *Automatic Text Processing; the Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, 1989.

[141] J. Sanghvi, G. Castellano, I. Leite, A. Pereira, P. W. McOwan, and A. Paiva. Automatic analysis of affective postures and body motion to detect engagement with a game companion. In *Human-Robot Interaction (HRI) 2011, 6th ACM/IEEE International Conference on*, pages 305–311, 2011.

[142] S. Schachter and J. Singer. Cognitive, social, and physiological determinants of emotional state. *Psychological review*, 69(5):379, 1962.

[143] R. C. Schank. *Dynamic memory: A theory of reminding and learning in computers and people*. Cambridge University Press, 1982.

[144] E. H. Shortliffe. *Computer-based medical consultations, MYCIN*. Elsevier Science Publishers, 1976.

[145] V. J. Shute and J. Psotka. Intelligent tutoring systems: Past, present, and future. Technical report, DTIC Document, 1994.

[146] D. Sleeman and J. S. Brown. *Intelligent tutoring systems*. Computer and people series. London Academic Press, 1982.

[147] C. A. Smith and P. C. Ellsworth. Patterns of cognitive appraisal in emotion. *Journal of personality and social psychology*, 48(4):813–838, 1985.

[148] R. P. Smith. Boredom: A review. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 23(3):329–340, 1981.

[149] P. E. Spector. Organizational frustration: A model and review of the literature. *Personnel Psychology*, 31(4):815–829, 1978.

[150] S. Srinivas, M. Bagadia, and A. Gupta. Mining information from tutor data to improve pedagogical content knowledge. In *Educational Data Mining*, 2010.

[151] M. Stern, J. Beck, and B. P. Woolf. Adaptation of problem presentation and feedback in an intelligent mathematics tutor. In *Intelligent tutoring systems*, pages 605–613. Springer, 1996.

[152] Y. Tong, W. Liao, and Q. Ji. Inferring facial action units with causal relations. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1623 – 1630, 2006.

[153] M. F. Valstar and M. Pantic. Fully automatic recognition of the temporal phases of facial actions. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 42(1):28–43, 2012.

[154] K. VanLehn, P. W. Jordan, C. P. Rosé, D. Bhembe, M. Böttner, A. Gaydos, M. Makatchev, U. Pappuswamy, M. Ringenberg, A. Roque, et al. The architecture of why2-atlas: A coach for qualitative physics essay writing. In *Intelligent tutoring systems*, pages 158–167. Springer, 2002.

[155] E. Verdú, L. M. Regueras, M. J. Verdú, J. P. De Castro, and M. Á. Pérez. An analysis of the research on adaptive learning: the next generation of e-learning. *WSEAS Transactions on Information Science and Applications*, 5(6):859–868, 2008.

[156] A. d. Vicente and H. Pain. Motivation diagnosis in intelligent tutoring systems. In *Proceedings of the 4th International Conference on Intelligent Tutoring Systems*, ITS '98, pages 86–95, 1998.

[157] E. Vockell. Educational psychology: A practical approach. *Purdue University*, 2004.

[158] J. J. Vogel-Walcutt, L. Fiorella, T. Carper, and S. Schatz. The definition, assessment, and mitigation of state boredom within educational settings: A comprehensive review. *Educational Psychology*

Review, 24(1):89–111, 2012.

[159] J. Wagner, J. Kim, and E. André. From physiological signals to emotions: Implementing and comparing selected methods for feature extraction and classification. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pages 940–943. IEEE, 2005.

[160] B. Weiner. An attributional theory of achievement motivation and emotion. *Psychological review*, 92(4):548–573, 1985.

[161] E. Wenger. Artificial intelligence and tutoring systems. *International Journal of Artificial Intelligence in Education*, 14:39–65, 2004.

[162] A. Wigfield and J. S. Eccles. Expectancy–value theory of achievement motivation. *Contemporary educational psychology*, 25(1):68–81, 2000.

[163] A. Wigfield and K. R. Wentzel. Introduction to motivation at school: Interventions that work. *Educational Psychologist*, 42(4):191–196, 2007.

[164] B. Woolf, W. Burleson, I. Arroyo, T. Dragon, D. Cooper, and R. W. Picard. Affect-Aware Tutors: Recognising and Responding to Student Affect. *Int. J. Learn. Technol.*, 4(3/4):129–164, 2009.

[165] Y. Yang. An evaluation of statistical approaches to text categorization. *Information retrieval*, 1(1):69–90, 1999.

[166] Z. Ying, Q. Bingzhe, and J. Chunzhao. On intelligent tutoring system. *CNKI Journal on Computer Science*, 04, 1995.

[167] Z. Zeng, M. Pantic, G. Roisman, and T. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):39–58, 2009.

Acknowledgement

First and foremost, I wish to express my utmost gratitude and heartfelt appreciation to my dedicated research advisors **Prof. Sridhar Iyer, Prof. Sahana Murthy, Prof. Campbell Wilson** and **Prof. Judy Sheard** for their guidance, continuous support and discussions throughout the course of my research work. Without my supervisors advice this thesis would not have been possible.

I thank **Prof. Sridhar Iyer** and **Prof. Sahana Murthy** for their encouragement, motivation and freedom they gave me to choose my research area and their guidance to achieve my goal. I thank **Prof. Judy Sheard** and **Prof. Campbell Wilson** for their support, guidance during my Monash stay and for their constant guidance for improving my thesis work.

I sincerely thank **Prof. Krithi Ramamritham** for his guidance, support during my initial days of research study and continuous guidance as my Research Progress Committee member. His comments helped me to explore the possibilities of a wider research domain and at the same time details to the finest granularity. I also take this opportunity to thank my other Research Progress Committee member **Ms. Janet Fraser** for her valuable comments that helped to improve my research work.

I also thank **Mr. Sridhar Rajagoplan**, CEO of Educational Initiatives, India, for his support and permission to use the Mindspark data for my research work. Also I would like to thank staff of Mindspark especially **Mr. Muntaquim** for his help in implementing my research work in Mindspark. I thank **Ms. Rajini Mitra**, teacher and school co-coordinator of Mindspark, for her support and extra effort she gave for my experiments. I am grateful to all the students who accepted to participate in my study. Their smiles made my experiments enjoyable.

I take the privilege to acknowledge my fellow researchers and facial expression observers **Aditi Kothhiyal, Anura Kenkre, Gargi Banerjee, Jagadish, Kirankumar Eranki, Neelamadhav** and **Yogendra Pal** for their help in facial observation coding and their support during my work. I thank my friend **Aditya Iyer** for helping us to learn and train ourselves for facial observation. I am grateful for his timely help and the time we spent together.

I am thankful to **Prof. Mohan Krishnamurthy**, CEO, IITB-Monash Research Academy, for his support, advice and guidance to handle tough situations during my research work. I thank all the members of Department of Computer Science, IIT Bombay and Caulfield school of IT, Monash University, who provided a friendly environment for research. My special thanks to the staff at IITB-Monash Research Academy, IIT Bombay, and Monash University. The most important names to be mentioned are **Mrs. Kuheli Mukerjee, Mr. Rahul Krishna, Mr. Rahul Ingle, Mr. Bharat Ingle, Mr. Adrian Gertler, Mr. Vijay Ambre, Mrs. Alpana Athavankar, Mrs. Sunanda Ghadge, Mrs. Subhadra, Mr. Thushar, Mrs. Gaikwad, and Mrs. Gayathri**. I thank **Ms Sheba Sanjay**, Communications and Training Services Consultant at IITB-Monash Research Academy, for doing the language review for my thesis.

My heart filled thanks to my lab-mates **Ajay Nagesh, Pratik Jawanpuria** in IIT Bombay and **Dwi A. P. Rahayu, Varsha Mamidi** in Monash University for their help and support during my work. I specially thank my friend **Naveen Nair** for his guidance, help and advice during my research. I am thankful to my friends **Anu Thomas** and **Chitti Babu** for their help and support, particularly during my stay at Monash University. I thank my hostel friends **Balamurali, Siddharth Gadkari,** and **Bharat Padeker** for their help and support during my stay in IIT Bombay. I duly thank my friends: **Ashutosh Namdeo, Asifkhan Shanavas, Kapil Kadam, Kunal Kumar, Ramesh K. V., Rupesh Bobade, Thaneshan Sapanathan, Wasim Feroze** and all my IIT Bombay campus friends.

I dedicate this thesis to my parents **Mr. Rajendran** and **Mrs. Vatchala Devi** who encouraged me to continue my education and for the support they provided during tough phases of my life. I thank my brother **Mr. Santhosh Kumar** and his wife **Mrs. Madhumathi** for their motivating words to take up big challenges in my life. I thank my grandma, grandpa, uncle and aunt for their care and affection during my early stages of life. I am grateful to all my family members especially **Krishanth**, **Vikranth** and **Lesanth** for their time and support.

I take privilege to thank people who helped and supported me directly or indirectly throughout my academic career. Last but not least, I thank the Almighty for giving me the strength and capability for this long journey in the pursuit of knowledge.

Date:
IIT Bombay, India **Ramkumar Rajendran**