A Machine Learning Approach to Modeling and Predicting Training Effectiveness

by

Alexander James Stimpson

B.S., University of Florida (2007)
S.M., Massachusetts Institute of Technology (2011)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2015

Signature redacted

Author .....
Department of Aeronautics and Astronautics
August 21, 2014

Certified by ..... Signature redacted
Mary L. Cummings
Visiting Professor of Aeronautics and Astronautics
Thesis Supervisor

Certified by ..... Signature redacted
Brian C. Williams
Associate Professor of Aeronautics and Astronautics and Computer Science and Electrical Engineering

Certified by ..... Signature redacted
William Long
Principal Research Associate, Computer Science and Electrical Engineering

Accepted by ..... Signature redacted .....
Paulo Lozano
Chairman, Department Committee on Graduate Theses

A Machine Learning Approach to Modeling and Predicting Training Effectiveness

by

Alexander James Stimpson

Submitted to the Department of Aeronautics and Astronautics
on August 21, 2014, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Developments in online and computer-based training (CBT) technologies have enabled improvements in efficiency, efficacy, and scalability of modern training programs. The use of computer-based methods in training programs allows for the collection of trainee assessment metrics at much higher levels of detail, providing new opportunities for training evaluation in these programs. These resulting datasets may provide increased opportunities for training evaluation and trainee intervention through the use of descriptive and predictive modeling. In particular, there is the potential for descriptive approaches to provide greater understanding of trainee behavior and indicate similarities between trainees, while accurate prediction models of future performance available early in a training program could help inform trainee intervention methods. However, traditional analysis techniques and human intuition are of limited use on so-called "big-data" environments, and one of the most promising areas to prepare for this influx of complex training data is the field of machine learning.

Thus, the objective of this thesis was to lay the foundations for the use of machine learning algorithms in computer-based training settings. First, a taxonomy of training domains was developed to identify typical properties of training data. Second, the theoretical and practical considerations between traditional machine learning applications and various training domains were identified and compared. This analysis identified the potential impacts of training data on machine learning performance and presented countermeasures to overcome some of the challenges associated with data from human training. Third, analyses of machine learning performance were conducted on datasets from two different training domains: a rule-based nuclear reactor CBT, and a knowledge-based classroom environment with online components. These analyses discussed the results of the machine learning algorithms with a particular focus on the usefulness of the model outputs for training evaluation. Additionally, the differences between machine learning applications to the two training domains were compared, providing a set of lessons for the future use of machine learning in training.

Several consistent themes emerged from these analyses that can inform both research and applied use of machine learning in training. On the tested datasets, simple machine learning algorithms provided similar performance to complex methods for both unsupervised and supervised learning, and have additional benefits for ease of interpretation by training supervisors. The availability of process-level assessment metrics generally pro-

vided little improvement over traditional summative metrics when available, but were able to make strong contributions when summative information was limited. In particular, process-level information was able to improve early prediction to inform trainee intervention for longer training programs, and was able to improve descriptive modeling of the data for shorter programs. The frequency with which process-level information is collected further allows for accurate predictions to be made earlier in the training program, which allow for greater certainty and earlier application of targeted interventions in a training program. These lessons provide the groundwork for the study of machine learning on training domain data, enabling the efficient use of new data opportunities in computer-based training programs.

Thesis Supervisor: Mary L. Cummings
Title: Visiting Professor of Aeronautics and Astronautics

Acknowledgments

I owe thanks to numerous people for the successful completion of this thesis.

First, to my advisor Missy Cummings, for believing in me and pushing me to achieve ever greater personal and professional growth. You were very instrumental and supportive through the qualifying exam process, and offered me fantastic opportunities in HAL. I will be forever grateful for your dedication to myself and the other students, and truly felt that you were always watching out for our well being both in times of success and struggle. With your help I have immensely matured as a researcher, professional, and human being.

To my committee members, Brian Williams and Bill Long. Thank you both for being willing to take on an unorthodox project that spans both human factors and computer science. Your insights have guided me along the way, and I am grateful for the time and advice that you have offered.

To my thesis readers, Joshua Downs and Matthew Cornick, for volunteering your valuable time to provide insights and suggestions that help to make this work successful.

To Jason Ryan, for all his help with edits and comments in the early drafts of this thesis. Your help has been critical to this work, and you've been a great friend who has provided support to me both in and out of the lab.

To Yves Boussemart, Jamie Macbeth, Luca Bertuccelli, and Kris Thornberg, for invaluable assistance in the research project, and help with research strategy, project and time management, and improving my writing skills.

To my UROPs, Hosea Siu and Scott Bezek for all your hard work on experimental setup, data collection, and coding. Your help was instrumental in the completion of the research contained in this thesis.

To Luisa Buinhas, for your help with preparing the Boeing CBT interface and collecting datasets for testing.

To Larry Young, for taking a risk on a graduate out of Florida and taking me under your wing. I will be forever grateful your generosity and guidance that has been fundamental in my educational and personal development.

To The Boeing Company, for providing both funding and support for my education and research.

To Sally Chapman, Liz Zotos, Beth Marois, and Marie Stuppard for all of your assistance, advice, and effort that you've provided to me. Through all my years in Aero-Astro, I will always recognize the tremendous support you offer to both the department and to the students.

To my many friends made during my time at MIT: Torin, Aaron, Jason, Kim, Justin, Allie, Jaime, Ashley, Luca, Yves, Kathleen, Mark, and many more. The times we have had together mave made my time at MIT more wonderful than I could have imagined.

To my Mom, Dad, and my brother Jon, for your unwavering love and support through my long time in school. You have always been wonderful for encouraging me to reach high and try hard. Your advice and encouragement have always given me drive to accomplish my dreams.

Finally, to my wife Jing. You've been with me through all the ups and downs I've encountered in the doctoral program, and have provided love, support, and encouragement every step of the way. You always know how to keep me motivated and kept pushing me to stay on track. I can never thank you enough for being by my side this whole time, and look forward to the rest of our lives together.

Contents

List of Figures

Q-1 Viewing a procedure as a sequence 349
Q-2 Example action sequence with transition counts and probabilities 353

List of Tables

Chapter 1

Introduction and Motivation

In its broadest sense, training can encompass a wide range of physical and mental endeavors to acquire knowledge, skills or competencies that are useful to a particular domain. Learning to play an instrument, practicing for a debate team, or learning to fly an airplane can all be characterized as forms of training. Training provides an organized and directed method for the introduction, acquisition, and retention of the required concepts and skills for the domain. From an industry perspective, training is usually driven by the need to increase performance by the employees, whether this be a new employee or refreshing the knowledge of a veteran employee. Training helps employees to hone their skills and reduce errors during their work, which can translate into increased efficiency and improved safety on the job. New employees may not be familiar with the particular systems, methods, or organizational elements at a new position, and it is standard practice to have an orientation or initial training period to ensure new personnel have the appropriate skills and knowledge to safely and efficiently perform their work [1–3]. Additionally, there is a large body of work that demonstrates that skills degrade with disuse [4–7], and since in almost every domain important skills may not be used regularly, organizations instead turn to training to maintain or improve performance in all facets of the job.

The modern ubiquity of computer use and internet access have dramatically impacted many facets of training. Many training programs are now incorporating computer-based training (CBT) or online elements. For example, more than 60% of the 5.7 million hours of training given by The Boeing Company in 2007 were partially or totally online, and this ratio is expected to continue to rise as the technology behind these systems

improves [8]. CBT training systems provide benefits beyond traditional classroom-based settings by allowing for greater numbers of trainees and providing more opportunities for trainee access to and engagement with the training material. This shift has been mirrored in the field of education, with a rapid rise over the last decade in the use of computer-based or online formats either to facilitate (e.g. online distribution of materials) or conduct higher-education courses. As of 2013 32% of all postsecondary school students enroll in at least one online course [9]. Primary and secondary schools have also begun to incorporate online and CBT learning programs into their curricula [10]. Online and computer-based environments provide new opportunities for gathering data on trainee behavior and performance both by allowing for the simplified recording of trainee inputs to the training system and by enabling more interactive assessment techniques. Additionally, these technologies can reduce the workload of trainers and allow for targeted training tailored to individual trainees. One distinct advantage of CBT is that data generated in these settings can be analyzed to provide additional feedback on individual trainees as well as about the training program itself.

1.1 Training Evaluation and Trainee Assessment

In any training program (both with and without computers), it is important for the effectiveness of the program to be measureable and monitored. Training programs can require considerable resources to implement, and often the organization will want to determine whether the time, cost, and effort put into a training program has met the desired objectives. The determination of whether or not a training program has met its goals is termed "training evaluation." The goals of a training program can be varied, including the acquisition of skills or knowledge by the trainees as well as the achievement of affective outcomes[1] such as improving trainee motivation [11]. If a training program fails to meet the intended goals or is not efficient in time or cost, it may be beneficial for the organization to modify the training program, either to increase the achievement of outcomes or to reduce costs. To properly make these decisions, it is important for the training evaluation methods to be timely and accurate. As with methods and metrics in other domains, if the training evaluation measure does not accurately represent the

[1] Affective outcomes relate to the moods, feelings, and attitudes of the trainee.

achievement of goals, it is difficult to use the measure to support organizational decision making processes.

A fundamental aspect of training evaluation is referred to in this thesis as "trainee assessment", which focuses on those metrics that identify whether individual trainees have acquired the desired skills and knowledge from the training material. These metrics can serve to assist the training evaluation process, as the acquisition of skills and knowledge will typically be an integral part of the goals of the training program. Additionally, these metrics can provide other benefits, such as supporting training intervention (TI). TI attempts to improve the acquisition and retention of skills by providing group or individually targeted changes to the standard training curriculum. Examples could include additional practice sessions or one-on-one lessons with the instructor (additional detail on TI methods is provided in Chapter 2). To determine whether it is appropriate to apply a TI methodology to a particular trainee, the supervisor must have information about individual performances.

Figure 1-1 presents a conceptual layout of a typical training program and associated training evaluation. As trainees complete modules of a training program, assessment metrics are gathered to determine the progress of each trainee. Based on these metrics, TI methods may be applied to assist struggling trainees during the training program. The individual trainee assessment metrics, along with program-level training metrics (such as percentage of trainees that pass the program), are compared with the program goals in the training evaluation process. The results of the training evaluation are given to the appropriate organizational elements (e.g. managers or other decision makers). If the evaluation indicates that the training program is not meeting its goals or that the costs outweigh the benefits, the organization may decide to implement changes to the training program.

The increased use of computers in training has dramatically impacted the availability of trainee assessment data. Computers can be used to collect information either remotely or on a more frequent basis (seen in online learning) or can create entirely new learning interfaces (such as CBT). In online learning, the types of assessments may not significantly change from traditional training environments, but are easily accessible by the trainees. Increased accessibility allows trainees to more frequently interact with the instruction material and thus allow the more frequent collection of assessment data.

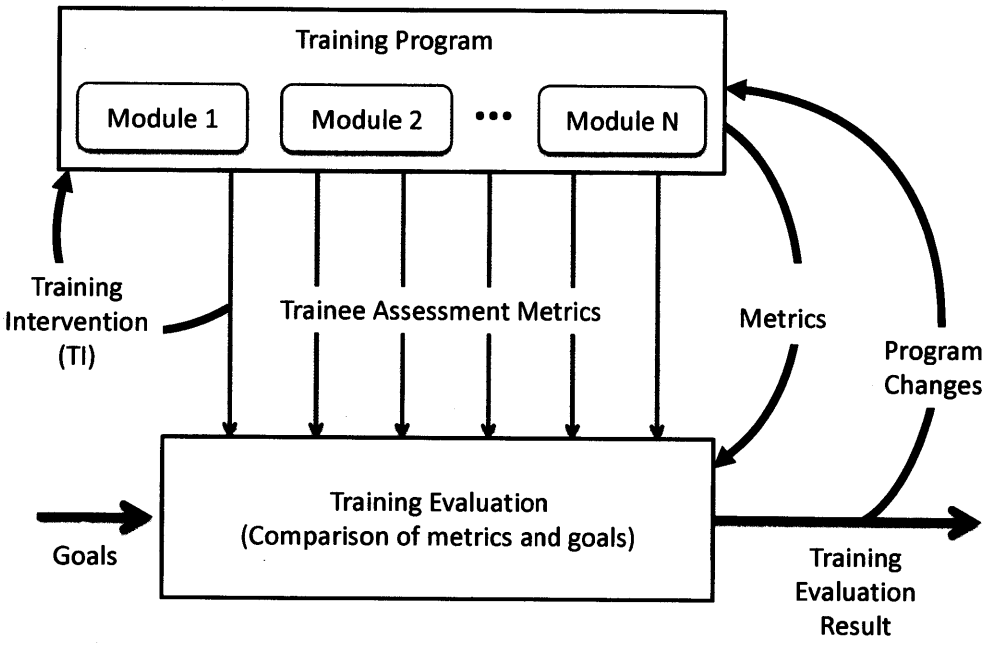


Figure 1-1: Conceptual representation of training program and evaluation.

CBT systems, on the other hand, allow for the development of new training strategies such as adaptive learning systems or the collection of detailed assessment information. Specifically, CBT training systems can log every interaction of the trainee with the system, providing detailed information that can be analyzed. Since many of these interactions will be sub-steps as part of the process of completing a greater objective, this thesis refers to this data as "process-level" information. Process-level information provides detail on not just *what* the current progress is in the current training module (e.g. the system state), but *how* the trainee reached that state. Not only can this information be used to identify the particular strategies that experts use in completing the training program, it can also diagnose particular behaviors and tendencies for error in individual trainees. While the potential benefits of process-level information are clear, collecting this information results in much larger datasets that are difficult to process by hand. To attempt to extract useful findings within this information, this thesis utilizes machine learning algorithms to assist in trainee assessment, training evaluation, and TI.

1.2 Machine Learning

Machine learning is a branch of artificial intelligence in which algorithms "learn" relationships in data. These algorithms can be applied in a predictive sense or to investigate internal relationships of a dataset. These algorithms utilize a database of past evidence to construct models that best explains the observations, mimicking human pattern recognition processes. For large or high-dimensional datasets, it is often difficult for a human analyst to develop models, whereas machine learning algorithms excel in this regime by being able to make use of the vast amounts of available data. These algorithms have been successfully applied in a wide range of applications, including computer vision, medicine, robotics, natural language processing, and search engines [12]. If there is limited availability of past evidence (i.e. a small dataset size), the models created may be inaccurate or inefficient.

Trainee assessment data from the use of online training and CBT systems fit well into the machine learning paradigm. Increased numbers of trainees (larger dataset size) and the availability of process-level information (higher dimensionality) potentially make machine learning a useful approach in extracting data from these systems. However, the suitability of applying machine learning in these environments and the impact of the training methodology on machine learning results must be investigated before the implementation of these algorithms as part of training evaluation or a targeted intervention approach.

1.3 Research Approach

Training can vary greatly in structure, methods, and assessment techniques, which may have important impacts on machine learning performance. Therefore, a taxonomy of training by task type was developed, utilizing the skills-rules-knowledge (SRK) framework developed by Rasmussen [13]. In this framework, training environments can be separated based on whether they train for a skill-based, rule-based, or knowledge-based task. The SRK structure indicates a generally increasing level of cognitive complexity (skills<rules<knowledge), of which the higher complexity environments (rule, knowledge) are of greatest interest for online and CBT settings. Typically, training methods and assessment data collected in each of these types of environments will differ from each

other, and the differences are discussed further in Chapter 2. Therefore, to assess the applications of machine learning in modern training environments utilizing online and CBT elements, this thesis focuses on data collected from the latter two of these training environments: a rule-based nuclear reactor CBT, and a knowledge-based classroom environment. For each setting, the suitability, advantages, and disadvantages of a range of machine learning algorithms are considered. Models are then constructed on each dataset using the most applicable techniques, and the results are analyzed both in terms of the model results and the implications for training evaluation and trainee assessment.

The first setting is a training environment for a procedure-based task, utilizing data collected from trainees on a simulated nuclear reactor CBT program. In this setting, trainees completed several training modules that familiarized them with the system and procedures before completing an assessment test module. The second training setting examined follows a more traditional classroom-style lecture approach, with data collected from a collegiate human-factors engineering course. The course included both theory and practical elements, and contains both traditional assessment methods such as classroom examinations as well as online components that allow investigations of process-level information. This training setting represents a training environment for a knowledge-based task that utilizes online interaction methods. Together, these two datasets are representative of markedly different training environments, and allow for the comparison of machine learning applications to utilize data generated from CBT and online training.

1.4 Research Questions

This thesis addresses several topics regarding the application of machine learning models to training data. The primary goals of the thesis are as follows:

1. Determine the typical properties of CBT data, how the training domain impacts the data obtained in these settings, and the considerations of applying machine learning algorithms to training data, with an emphasis on procedure-based and rule-based training environments

2. Assess the usefulness of supervised and unsupervised learning algorithms in example training datasets from different types of training domains, particularly from the perspective of a training evaluator.

3. Determine the importance of the detailed temporal and process-level information provided by online and CBT formats in the application of machine learning to the data gathered from the example training environments

1.5 Expected Contributions

In answering the research question above, this work presents a series of contributions to training, training evaluation and machine learning on datasets generated in human training programs. These are fully discussed in Chapter 7, and are briefly summarized here:

1. The development of a taxonomy of training domains based on the training task based on the SRK framework

2. A comparison and selection of metrics for the measurement of procedure adherence in procedure-based task environments

3. The identification of appropriate data preparation techniques prior to the application of machine learning algorithms to human training data, including dimensionality reduction

4. The comparison and selection of appropriate machine learning techniques for unsupervised and supervised learning on datasets from both rule-based and knowledge-based training environments

5. The identification of how the machine learning results on the example datasets could be utilized by training supervisors and evaluators to both improve the performance of the trainees and provide feedback for the training program

6. The comparison of the potential contribution of the more detailed "process-level" information to traditional assessment metrics for informing training evaluation and TI when used as part of a machine learning approach

1.6 Thesis Organization

The dissertation is organized as follows:

- Chapter 1, *Introduction and Motivation*, presents the motivation, approach, and goals for the research.

- Chapter 2, *Background*, presents a review of training, training assessment, and machine learning literature to inform the discussions of the application of machine learning to training data.

- Chapter 3, *Machine Learning Considerations for Training Applications*, discusses the different types of training, the differences in data provided by these types, and how these differences impact a machine learning approach.

- Chapter 4, *Machine Learning in Rule-Based Training Environments*, analyzes the effectiveness of machine learning algorithms in an exemplary procedure-based CBT dataset to inform training evaluation and trainee assessment. Chapter 4 also presents the CBT interface used and a discussion of the procedures and training program utilized.

- Chapter 5, *Machine Learning in Knowledge-Based Training Environments*, performs similar analysis as Chapter 4 on an exemplary knowledge-based classroom dataset with online elements.

- Chapter 6, *Model Comparison*, compares and contrasts the effectiveness of the machine learning models in each of the two exemplary training domains, and presents rationales for the results.

- Chapter 7, *Summary and Conclusion*, summarizes the findings in the thesis, and provides a set of general conclusions and recommendations for future research.

Chapter 2

Background

This thesis focuses on the use of machine learning models in training evaluation, trainee assessment, and trainee intervention (TI). Developments in CBT and online training technologies have enabled the availability of large, process-level datasets, and the use of machine learning approaches may have considerable advantages for informing training evaluation and TI on these datasets. To provide background to support this research, this chapter is divided into three main sections that summarize prior work in relevant research areas. The first presents an overview of training and training assessment. It also highlights the advancements in some of the major applications of training research, including medicine, aviation, and the military. The second section of this chapter presents an overview of machine learning methods in both unsupervised and supervised approaches. The third section provides a discussion of the field of Educational Data Mining (EDM), which covers some machine learning approaches in the field of education. In each of these sections, background information that is particularly relevant to machine learning approaches in training is highlighted.

2.1 Training and Training Assessment

Training is an integral part of virtually every profession. Whether a person is working as a car mechanic or a nuclear reactor operator, training is necessary to prepare them to perform the duties of the job. As many fields utilize increasingly complex technology, training on the use of this technology becomes progressively more important. Herein training is defined as "organized activity aimed at imparting information and/or instruc-

tions to improve the recipient's performance or to help him or her attain a required level of knowledge or skill" [14]. The increase in performance may refer to training for a skill or environment, where the trainee has not been previously exposed to the training subject. Examples could include training a new employee for the duties of a job, or a current employee being trained on a new software interface. Training can also refer to refresher training, where the trainee has already undergone training on the subject. In many safety-critical fields, refresher training aims to counteract the noted gradual decrease in performance over time [5, 7].

The overarching goal of training is to yield knowledge and skills that are useful, durable, and flexible, with the intention of improving the performance of the trainee [15]. These goals focus on the long-term, and extend well beyond the duration of the training itself. The acquisition of skills and knowledge has been widely studied from both a theoretical and practical standpoint (e.g. [5, 7, 16–22]). The need for training evaluation has also been noted in a variety of works (e.g. [11, 15, 23–34]). The following subsections summarize the past and present theories and applications of training and training evaluation.

2.1.1 Training

From a practical standpoint, training has been an instrumental part of society for centuries, often in the form of apprenticeship. More recently, scholars have been interested in investigating the details of the physical, psychological, and cognitive aspects of training. In his pioneering book *Hereditary Genius* [35] in 1869, Sir Francis Galton recognized the need for training in order for an individual to achieve maximum potential in a field. He also indicated that early performance gains were rapid but diminished with the amount of training, otherwise known as the learning curve. According to Galton, a trainee's maximum potential was limited fundamentally based on the innate skills of the individual, a perspective which remains in more contemporary theories of skill acquisition [21, 22].

Figure 2-1 shows a notional graph of the improvement in performance with experience. Expert performance requires both the development of cognitive and associative skills, as well as a high level of experience. Without full development of cognitive skills, the trainee may not reach true expert performance, termed "arrested development" in Figure 2-1. For skills that are heavily practiced but do not require significant cognitive input, auto-

maticity of the skill may be developed. In this phase, little or no attention or cognitive effort is required to perform the skill [22]. The literature discusses the differences between "procedural" and "declarative" training, which generally separates training by the cognitive complexity of the associated skills [26,36]. This indicates that there are fundamental differences between the nature and development of skill acquisition dependent upon the cognitive complexity of the task. While there has been much research on the general acquisition of skills (e.g. [16,20,37,38]), this area is beyond the scope of this thesis.

2.1.2 Computer-Based Training

Modern training programs have begun to utilize technological advances in CBT and online training to improve the efficiency and effectiveness of the training program [39–41]. Early research on CBT focused on the comparisons between training through CBT and traditional learning environments, particularly in military and educational settings [42,

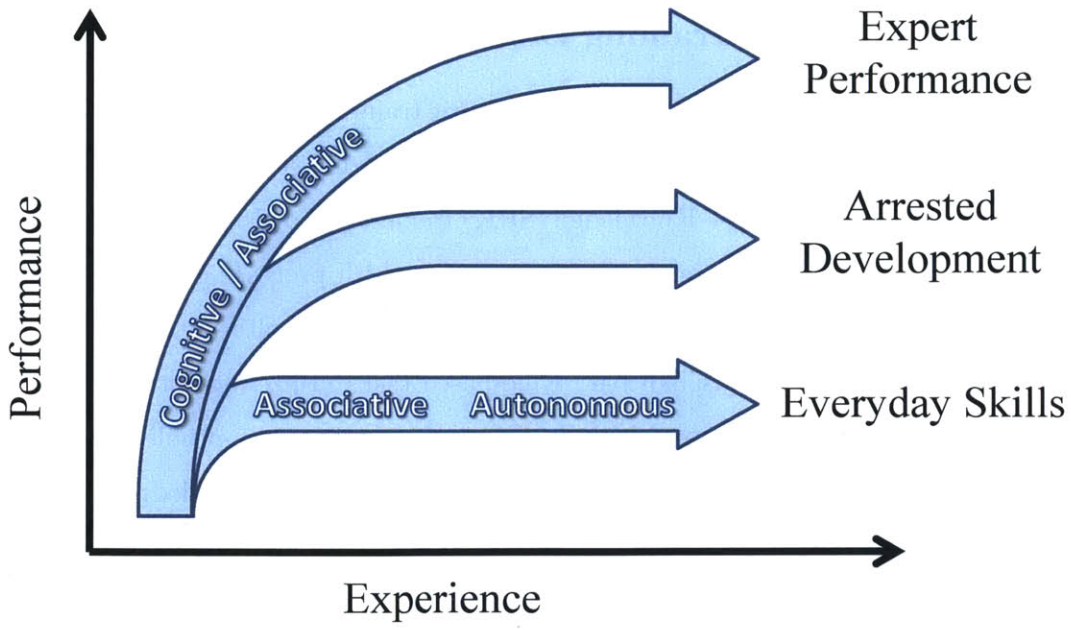


Figure 2-1: Notional diagram showing development of performance with experience. Adapted from [16].

43]. Similar studies have since been carried out in other domains such as medicine [44–46] and vehicle operation [47,48]. Overall, these studies have pointed to reduced instruction time and increased cost savings provided by CBT as compared to traditional instruction methods, but generally do not support increased learning or training transfer through the use of CBT methods over traditional approaches [41]. More recently, research on CBT has focused on the structure and design of the training environments to maximize learning and trainee acceptance [39, 40, 49–51]. However, there is little discussion in the CBT literature of the differences in advantages of computer-based methods between procedural and declarative training settings.

Research on CBT has indicated advantages in training program efficiency and scalability which will continue to encourage the increased use of these methodologies, though the need for increased evaluation of CBT programs is widely reported [41,52,53]. However, for all training (CBT or otherwise) the methods, assessment data and evaluation techniques may vary widely dependent upon the domain. An exhaustive list of training domains is not necessary here, but it is helpful to define a general taxonomy of training domains.

2.1.3 Taxonomy of Training Domains

There are an immense number of domains that utilize training to improve the performance of the trainee. These can be divided according to the primary types of tasks involved using Rasmussen's Skills-Rules-Knowledge (SRK) framework to classify domains by task type [13]. Briefly, skill-based tasks refer to those for which processing is automatic and often require little conscious effort [54]. Examples include monitoring an interface for some change or swinging a baseball bat. Rule-based tasks utilize heuristics to limit the cognitive effort needed. Examples include usage of procedures or problem diagnostics such as an operator investigating an error in a nuclear reactor. Knowledge-based tasks require the greatest amount of cognitive effort, and may have multiple goals or subgoals that often require assessing open-ended problems, such as completing a physics exam. Each of these types of tasks requires different proficiencies, and training methods and objectives will be different for each of these domains.

The latter two groupings (rules and knowledge) overlap well with previous discussion of the importance of considering the differences between "procedural" and "declarative"

objectives for training [26, 36]. It is easy to see why the differences between the SRK domains are important from a training perspective; consider the methods used for training an athlete, instructing an operator for a nuclear power plant, or teaching aerodynamics to a student. Athletic training falls primarily under skill training, nuclear reactor operators work in a highly procedural rule-based environment, and learning aerodynamics is a knowledge-based task that requires the development of abstract cognitive models. Table 2.1 presents a summarized list of the training considerations in each of the domain types from the SRK framework.

Table 2.1: Considerations for SRK domains

Task Type	**Desired Outcomes**	**Typical Training Characteristics**
Skills	Automaticity of actions, development of muscle memory	High use of repetitions, action-focused, sensorimotor training
Rules	Adherence to standard protocols, development of heuristics, familiarity with diagnostic techniques, context awareness	High use of repetitions, procedural training, operational simulations
Knowledge	Development of conceptual understanding, use of analytical techniques, ability to extrapolate skills to novel situations	Classroom-based lecture format, no repetition of assessment questions

In a real-world setting, training may involve multiple types under the SRK framework. For example, consider training a nuclear reactor operator. There are many procedures that must be memorized and followed to maintain safety, which would fall under "rules." However, it is also critical for the operator to have a general understanding of the reactor system as a whole and the basic science behind the reactor's operation, which would fall under "knowledge" training. While this would seem to complicate the division of training and evaluation methods based on the SRK framework, in fact at its elemental level each task still represents training for a specific domain (e.g. "knowledge"). In this sense, these tasks represent building blocks for the operational environment.

In the discussion of specific training domains, it is important to consider not only the typical training methods but also the training evaluation strategies relevant to the domain. The next section presents background in training evaluation as preparation for

a discussion of specific training domains.

2.1.4 Training Evaluation

Training evaluation is an integral part of most training programs. It represents the set of tools and methodologies for assessing the effectiveness of the training program [23], and can include metrics from both during and after the completion of the training program (such as performance in the operational environment). Without this type of measurement, it is impossible to know whether the training program has met the desired goals, such as improving the performance of the trainees. Evidence of meeting the training goals has many uses, such as demonstrating the value of the training program [25,55–57]. There are six general approaches to training evaluation: goal-based evaluation, goal-free evaluation, responsive evaluation, systems evaluation, professional review, and quasi-legal [58,59]. Of these, goal-based evaluation and systems evaluation are the most common [23], and will be briefly discussed here.

Training Evaluation Models

The most recognized methodology for training evaluation is goal-based, described by Kirkpatrick [60] as having four main levels of evaluation: 1) reactions, 2) learning, 3) behavior, and 4) results. The first level, reactions, focuses on the reaction of the student to the training, which influences how much can be learned [25]. The second level focuses on measuring the evidence of learning, and thus the possibility for changes in operational behavior. Kraiger et al. proposed a taxonomy for learning outcomes that divides outcomes into cognitive, skill-based, and affective outcomes [11], as shown in Figure 2-2. The third level, behavior, focuses on the transfer of knowledge and skills learned in training to the operational environment, also called "training transfer" or "transfer performance". Figure 2-3 presents a widely recognized model of training transfer provided by Ford and Weissbein [27]. The last level, results, measures the organizational impact of the training program. With increasing levels of Kirkpatrick's typology, the information provides more detail on the impact of the training program. However, it also represents typically greater difficulty in measurement. The first level can be measured fairly simply with surveys or other reporting mechanisms at the time of training, while the fourth level might involve an organization-wide investigation. Perhaps as a result of the increasing difficulty, it is

widely reported that organizations typically evaluate training only at the lower levels of Kirkpatrick's typology. The American Society for Training and Development (ASTD), indicates that from a set of organizations surveyed, the frequency of measurement of each of Kirkpatrick's levels are 93%, 52%, 31%, and 28% for levels 1, 2, 3, and 4, respectively [23], with similar statistics reported for Canadian organizations [25]. The lack of evaluation at the higher levels of Kirkpatrick's typology, and the need for increased evaluation of training are widely reported in academic literature [23, 26, 61–66].

Several extensions and modified versions of Kirkpatrick's typology have been introduced more recently. Tannenbaum et al. added post-training attitudes to the model, and specified a division of the behavior level into training performance and transfer performance [67]. This model formally defines the difference between training and transfer performance, which relates to the greater issue of the difference between performance and learning, which has been widely reported in the literature [15, 33, 65, 68–72]. At its heart, this issue recognizes that strong performance in training does not necessarily translate

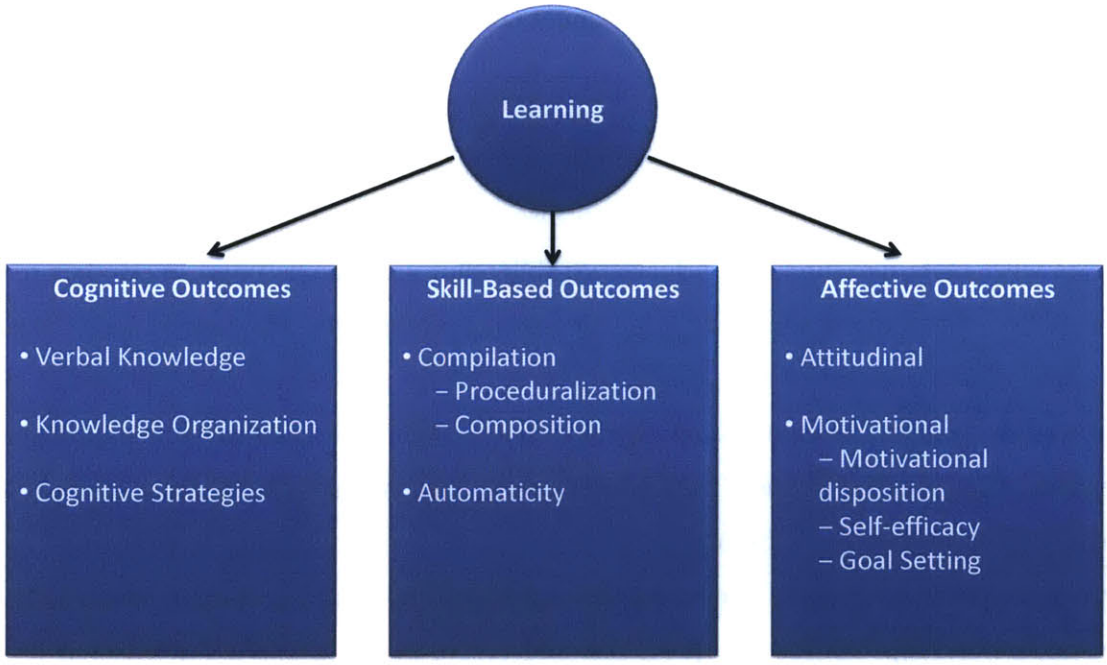


Figure 2-2: Classification scheme for learning outcomes. Adapted from [11].

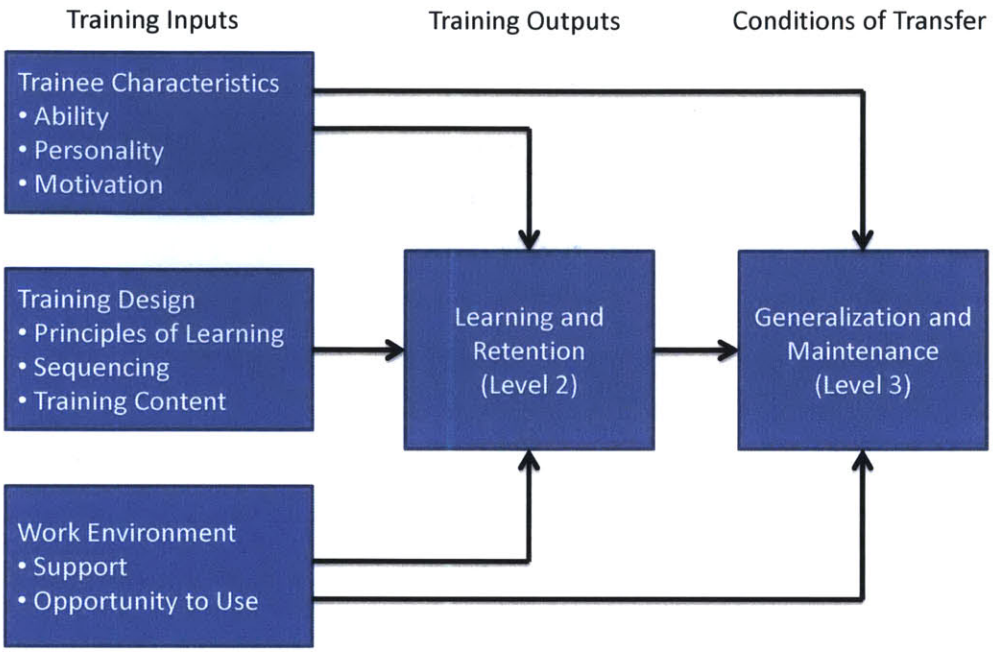


Figure 2-3: A model of training transfer. Adapted from [27].

into strong operational performance. That is, a trainee may be able to perform well on the training task, without any real learning taking place that would result in differences in behavior in the field. The reverse can also be true, that learning can occur during times with little measureable performance difference, as demonstrated by studies from the 1930s-1950s in latent learning [73] and motor learning [74]. The association of training performance with learning creates misinterpretations of errors during training and testing of trainees. Training programs may try to minimize errors in training, without realizing that the improvement in training performance may not transfer into operational performance [65]. Mottos from the military such as "we do it right the first time" and "we don't practice mistakes" demonstrate misunderstandings of errors in training. These misunderstandings can also be prevalent and create issues at the organizational level; instructors that are also evaluated by their trainees' performance may specifically try to minimize errors during training and testing (the so-called "teaching to the test") [65]. Unfortunately, many organizations do not evaluate transfer of training (the behavioral

level of Kirkpatrick's typology) or use measures that lack validity [65,66].

Other extensions of Kirkpatrick's model include those from Holton [75,76] and Kraiger [77]. In Holton's model, the level of reactions is removed and links are added between several trainee characteristics and training and transfer performance. Kraiger's model emphasizes training content and design, changes in learners, and organizational payoffs. While these newer models attempt to update aspects of Kirkpatrick's original typology, most academic and organizational sources still utilize the original version for training evaluation. Despite the common usage of the Kirkpatrick typology in training evaluation, it is widely reported that advances in training technology (such as online and CBT settings) require better evaluation methods not only after training but during training itself [28,32].

Some other commonly used evaluation methodologies are system-based, including Context, Input, Process, Product (CIPP) [59], Input, Process, Output, Outcomes (IPO) [78], and Training Validation System (TVS) models [23, 79]. These models help the evaluator to better consider the contextual and situational aspects of the training as compared to Kirkpatrick's goal-based model but generally do not represent interactions between training design and evaluation. They also do not provide detailed descriptions of the processes behind each outcome and do not provide tools or recommendations on how to conduct the evaluation. As such, the implementation of these models varies considerably with a lack of comparability across studies.

General Evaluation Considerations

There are several important considerations that must be made in the evaluation of any training program. First, the training domain will influence the information and skills to be learned and the overall goals of the training program. Second, the particular training methodologies, such as classroom lectures, CBT programs, or simulations will dramatically change the data available for evaluation. Different data types and the granularity afforded by different methodologies will impact the evaluation strategy, as will be seen in Chapter 4 and 5. Often, the methodologies used will also be dependent upon the particular domain. Third, the background and attitudes of the trainees must be considered in the evaluation, in addition to the objective and subjective performance measures. In particular, there may be large differences between in evaluation between training and

re-training programs, as exhibited through the differences between novice and expert trainees [19,80,81]. An example of the difference in behavior is that experts tend to skip steps or modify procedures to improve completion efficiency; training evaluation should account for these behavioral differences.

Training evaluation must also consider the impact of the evaluation itself on the trainee [82–88]. Trainees will often determine how they are being assessed and will modify their behavior to match that assessment in an attempt to improve performance. While this may raise their training performance, it may not transfer into the job, as noted previously. Thus, any evaluation method must consider the impact of evaluation on the trainees. It has been proposed that the tendency for trainees to adapt to training examinations can be utilized through judicious design of the evaluation method, such that the evaluations focus on aspects that have strong transfer into the field [89]. Due to the importance of the training domain in training evaluation, several examples of important domains in training are presented in the following section.

2.1.5 Training Domains

Since most practical application of training and training evaluation comes from particular domains, it is worth highlighting important results from the more heavily researched domains. The following sections briefly discuss training and training evaluation applications to commonly reported domains, including medicine, aviation, and the military.

Training in Medicine

Prior to World War II, medical competence was primarily assessed using an apprenticeship model, where a mentor would be solely responsible for subjectively determining when a medical trainee was adequately prepared [80]. As the number of medical trainees rose, this was no longer logistically practical and led to new methods of trainee assessment including Multiple Choice Questions (MCQs), written simulations including the Patient Management Problem (PMP) [90] and the Modified Essay Question (MEQ) [91, 92], learning process measures such as the Triple Jump Exercise [80], and live simulations such as the Objective Structured Clinical Examinations (OSCEs) [80,93–96], which have become common in modern medical training [97,98].

MEQs were the solution to the logistical issues associated with the apprenticeship

model by being scalable to higher numbers of trainees and efficient to score, and have been supported by the use of computers in modern training programs. However, they have been challenged for not requiring the active generation of responses, only requiring the trainee to either identify the correct answer or eliminate the incorrect ones [99–101].

Written simulations such as the PMP and MEQ attempt to measure clinical reasoning, and sometimes allow branching solution paths or require the trainee to collect data on the patient [102]. However, these methods have been criticized for being very specific to the case used in the simulation [103], along with issues of disagreements between evaluators on the correct pathways and scoring of trainees [104, 105]. To try to reduce the dependence on the case selected, several research groups have suggested a "key feature approach" that only focuses on the key elements of the simulation rather than every action [106, 107].

Learning process measures recognize that an important aspect of training evaluation lies within the learning process, which was introduced in Chapter 1 as "process-level measures." While some researchers have recognized the importance of this information, it has largely been ignored in implementation within medical training programs [80]. Live simulations such as OSCEs are currently the most common evaluation methodology. These involve trainees rotating through a series of 10-20 simulated cases, collecting patient information and determining and executing the appropriate actions. By allowing for many cases in a single simulation, it helps to remove the issue of case specificity seen in written simulations. These simulations attempt to measure both the procedural skills of the trainee as they perform the necessary steps, as well as the declarative knowledge associated with diagnosing the illness and identifying the correct treatment strategy.

Most of these assessment methods still include subjective rating systems that comprise either part or all of the final assessment metrics. This may be partly an artifact of the apprenticeship traditions of the profession, or also due to the relative ease with which ratings can be gathered compared to objective sources. However, these ratings have been widely criticized due to their lack of accuracy and reliability [98, 108, 109], and it has been demonstrated that there is little consistency both between and within raters [110–113]. It is clear that there continues to be a push within the medical training community to move to more objective measures of trainee performance. Additionally, medical training environments require both procedural knowledge of the treatment steps and declarative understanding of the symptoms, diagnoses, and treatments. The greater use of computers

in modern training programs as part of objective assessment methodologies such as MCQs and OSCEs have dramatically increased the availability of process-level information, and this domain is a likely candidate for the use of machine learning algorithms in trainee evaluation.

Training in Aviation

There are a wide range of roles in aviation that require training, including pilots (or other operators), maintenance crew, air traffic control (ATC), among others. The Federal Aviation Administration (FAA) provides guidance on a range of training for aviation personnel, including Crew Resource Management (CRM) training, Maintenance Resource Management (MRM) training, electrical systems training, and even the design and implementation of operational simulations for training [114–117]. For pilots or other operators, simulation remains the most common training method; flight simulators can be highly capable, including features such as motion platforms and out-the-window views [118,119]. These simulators allow pilots to practice interacting with the aircraft controls in a wide range of training scenarios. Evaluation of trainee performance holds many parallels with medical training, with both objective simulator measurements and subjective self-rating and expert performance rating utilized as evaluation metrics.

Aviation is another field that requires the use of both procedural and declarative skills. Aviation environments typically make frequent use of checklists and other procedures both in the operation and maintenance of aircraft. Additionally, it is important for operators such as pilots to understand the fundamental physical properties that govern their vehicle, and training programs will include declarative knowledge elements such as courses in aerodynamics and aircraft systems (e.g. [120]).

In recent years, a particular emphasis has been placed on CRM training as a field of research [29,32,121]. CRM training is a form of team training that focuses on the interactions between personnel in the cockpit. Topics typically include teamwork, leadership, situation awareness, decision making, communication, and personal limitations [29,121,122]. A general review of CRM training programs shows mixed results, with most evidence of learning coming from changing self-reported attitudes [121]. Evaluations of CRM training report difficulties in separating behavioral changes due to flight training from those due to team training. This difficulty may be encountered in other forms of team training,

where performance metrics are dependent upon learning from both individual and team training. Despite the challenges, there is a clear need identified for additional research and development of objective team training methods.

Training in the Military

Training in the military has been an important part of ensuring an effective military force for millennia. Often this training includes both mental and physical aspects and can cover all areas of the SRK framework. The full range of training programs implemented by the US and other militaries is too extensive to present here, however there are several important aspects of military training that are worth noting. First, military training programs have readily incorporated new developments in training technologies such as CBT and advanced simulation environments [119]. This indicates that new datasets are becoming available within military training settings that are conducive to the application of machine learning algorithms. In fact, basic statistical learning algorithms have been previously used on subjective training data for prediction of pilot performance [123]. However, the clustering and prediction methods were only used on ratings of cognitive concepts, and not on performance data such as would be available from more modern computer-based simulation training programs. An important result of this analysis was the identification of the usefulness of machine learning methods for trainee selection and evaluation in military domains. Second, military training programs will often consider more unusual training program design and implementation strategies compared to other organizations [124]. Military organizations have investigated learning during sleep, accelerated learning programs such as Event-Based Approach to Training (EBAT), Suggestive Accelerative Learning and Teaching Techniques (SALTT), neurolinguistic programming (NLP), and paraspychology as part of training programs [124, 125]. While there may be little scientific evidence to support many of these concepts [124], it remains an indication that military training programs may be the most willing to change and adapt to novel training techniques.

2.1.6 Training Summary

This section has provided an overview of research in training and training evaluation, and presented some of the differences and similarities across training domains. Some

important highlights for the consideration of the use of machine learning algorithms in training evaluation are summarized here:

- Training design, implementation, and evaluation may vary dramatically based on the task type (e.g. Skills, Rules, Knowledge)
- There remains a large dependence upon subjective rating systems in training evaluation, which have been challenged for their reliability and validity
- Evaluation methods should consider not only in-training performance, but attempt to measure the learning (training transfer) that occurs during the training process
- Models of training performance and training transfer have the potential to assist with training evaluation, trainee selection, and training design and implementation

With a general knowledge of the common design, implementation, and evaluation principles, we can now consider the available machine learning models and their appropriateness to training. The following section introduces the field of machine learning and the algorithms used for testing on training datasets in this thesis.

2.2 Machine Learning

Machine learning (or data mining) is a branch of artificial intelligence that focuses on algorithms that "learn" the relationships between data. These relationships can be descriptive by providing greater understanding of the inherent properties of the data (unsupervised learning), or predictive by being able to estimate values for new data (supervised learning). These algorithms have risen in popularity in recent years, which likely can be attributed in part to their flexibility and capacity to process large, high-dimensional datasets [126]. A well-recognized formal definition of machine learning from Mitchell is: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E" [127]. It is important to note that these algorithms are not learning in the human sense, but able to construct better models the more data (experience) provided to the model.

In this field, algorithms incorporate a set of previously obtained data, called "training data," to improve the performance of the algorithm. Training data is used to set parameters in the machine learning models, resulting in the construction of a model that can then be tested. Model performance is usually measured on a set of data that was not used for training, and often referred to as "test data." It is important to distinguish between data collected from training domains discussed earlier and training data, which is used to train machine learning models.

An important aspect of any machine learning approach is referred to as "feature selection." Consider a model of basketball player skill. A wide range of data may have been collected that includes their height, age, vertical leap, and whether or not they were afraid of monsters as a child. As one can imagine, some of these variables may be more useful in modeling basketball player skill than others. The determination of which variables represent important and useful quantities for implementation in the learning algorithm is feature selection, where the variables are referred to as "features." Selecting strong features can make a learning algorithm efficient and provide strong prediction performance, while weak features will result in models that are overly complex or have poor prediction performance. Feature selection is an important part of any learning algorithm; choosing a more powerful learning technique will not totally overcome a poor selection of features.

Machine learning techniques typically make several assumptions about the data used to train and test the algorithm. The algorithms will result in the best model performance (descriptive or predictive) only if these assumptions are met. Some of the major assumptions are as follows:

- Large amount of data – most machine learning algorithms rely on having a large number of training examples for the algorithm to learn
- Large number of available features – having sufficiently large feature space will allow feature selection to identify useful features to use in the models
- Well-characterized noise – like most models, machine learning algorithms assume that data is generated by a consistent process, and that variations in the data follow a pattern

Dependent upon the particular dataset and the way in which the data is generated,

it may or may not meet these assumptions. While machine learning algorithms can be trained in the absence of one or more of these assumptions, the results may be unpredictable. Understanding how machine learning algorithms behave on atypical datasets is an important area of research, and determining how datasets from training programs fit within the standard machine learning paradigm is one of the main focuses of this work.

In the human training domain, it is difficult to meet the standard assumptions of most machine learning algorithms given above. Typically, only a relatively limited number of trainees are available for data collection, and the variations in performance may change as the trainee learns over the course of the program. A more detailed list of the differences is shown in Table 2.2.

Table 2.2: Differences between human training and traditional machine learning datasets

	Human Training	**Traditional Machine Learning Domain**
Data Points[1]	Data based on number of trainees, typically in the 100s to 1,000s	Database of 10,000s-1,000,000s of values
Features	Dependent on the number of evaluation measures, often <10 per training topic	Characteristics typically 100s-1000s of possible features
Noise	High level of human variability, will change as trainees gain expertise in the subject matter	Consistent noise such as a fixed probability distribution

Different machine learning methods will have various robustness to datasets dependent upon the properties (data points, features, noise) shown in Table 2.2; that is, the performance of some algorithms will be impacted more by these properties than others. One of the goals of this thesis is to investigate the ability of different machine learning algorithms to operate under the characteristics of human training data. Before discussing the specific applications of machine learning algorithms to training, it is important to understand the general categories of machine learning algorithms and their uses in other

[1]The term "data point" used in this thesis refers to an individual instance or observation in the data.

domains. The diverse set of machine learning algorithms can be divided into two main groups: unsupervised learning and supervised learning. Each of these groups utilizes different approaches and targets different goals.

2.2.1 Unsupervised Learning

Unsupervised machine learning focuses on finding inherent relationships within a dataset based on the features provided to the algorithm. These algorithms look for commonalities or patterns in the data itself. The two most common types of unsupervised learning are clustering and dimensionality reduction methods.

Cluster Analysis

Clustering algorithms find groupings, or "clusters" within the data. Identifying clusters in data can be useful in helping to identify particular points in the data that show similar feature values to each other. When applied to a training setting, the features correspond to particular performance scores or even individual actions. In essence then, cluster analysis can be used to find trainees that exhibit similar behaviors or performance profiles. This can be useful in an assessment and intervention setting, as it is able to clearly identify a set of trainees with similar behaviors that could be combined for a group intervention strategy. However, there are several considerations that apply across all clustering algorithms that must be taken into account before implementing the algorithms on a dataset. First, different algorithms exhibit different strengths and weaknesses in the clustering approach. Second, transformations or standardization of the data may be required as cluster algorithms are highly dependent upon measuring distances in the feature space. Third, a particular distance measure itself must be selected for the dataset to determine similarity between points in the data. Once these have been addressed, cluster algorithm performance may be compared for the dataset. The following sections outline the algorithms and strategies for cluster analysis.

Clustering Algorithms

There are a wide range of algorithms that have been developed for clustering datasets. While an exhaustive discussion of all of the clustering algorithms is not possible here,

this thesis presents four of the most common clustering methods: k-means, agglomerative clustering, Gaussian Mixture Model (GMM), and Self-Organizing Map (SOM). Each method represents a common form of one of the main types of clustering algorithms: centroid models, connectivity models, distribution models, and topological models, respectively. The remainder of this section introduces these algorithms in greater detail, along with outlining the strengths and weaknesses of each algorithm.

The k-means algorithm is generally considered the most popular clustering algorithm. It solves a specific optimization problem of minimizing the sum of squared distances between data points and the cluster centroid to which they are assigned. Most commonly it is calculated in an iterative two-step approach that first updates the cluster centroids and then reassigns data points to the closest cluster. It can be thought of as a relaxation of cluster centroids into a local minimum based on the location of nearby data points. It is a simple and efficient algorithm, which makes it desirable on large or complex datasets. However, it does require the user to pre-specify the number of clusters, and determining the optimal number of clusters typically requires the iterative creation of models and comparison through criteria such as the Bayesian Information Criterion (BIC). k-means is also sensitive to outliers, as it will heavily penalize the presence of data points at a far distance from the cluster centroid. Finding the global optimum of cluster centers and cluster assignments has been identified to be NP-hard (from computational complexity theory). Briefly, this indicates that algorithms guaranteed to find a solution typically require high computational effort to solve, and faster algorithms (such as heuristic approaches) are not guaranteed to find the optimal solution. To avoid the difficult computation in finding the global optimum, a greedy approach is taken for adjusting the cluster centers and thus any particular run of the k-means algorithm is understood to find a local optimum. Despite these drawbacks, the simplicity and efficiency of k-means ensure that it remains a popular selection for clustering.

Agglomerative clustering is a form of hierarchical clustering. It generates a hierarchical tree of nodes representing clusters, where the lower clusters on the tree are members of the higher clusters. In this way, a map of cluster assignments is created from the top level where all data points are in a single cluster, to the bottom level where each data point is in its own cluster. To create this tree, agglomerative clustering begins at the bottom, combining the closest clusters until only one cluster exists at the top of the tree. Since

this method requires the combination of clusters that are "near" each other, a distance parameter known as the "linkage criterion" must be selected. Examples include single-linkage clustering, which measures cluster distance by the two closest points from the clusters, or average linkage clustering which averages all the distances between each pair of points between the two clusters. Like k-means, agglomerative clustering is simple to implement and easy to interpret due to the hierarchical tree created and can be visualized through dendrograms. It also calculates cluster splits for all levels of the tree at once; to select a specific number of clusters the user merely needs to "cut" the tree at the desired number of nodes. However, it is of high complexity, being $O(n^3)$ in the general case (where n is the number of data points to be clustered), though it can be shown to be of $O(n^2)$ in particular special cases [12]. Like k-means, hierarchical clustering also encounters difficulties clustering outliers.

Gaussian mixture models (GMMs) take a distribution approach to clustering. The user selects a certain number of multivarate Gaussian distributions which are fit to the data. In this case, each distribution represents a cluster, with points being assigned to the cluster with the highest probability. Since Gaussian distributions are continuous over the entire feature space, it provides easy calculation of membership for new points, and easily fits into "soft clustering" approaches where a point has partial membership to multiple clusters. It also captures correlation and attribute dependence in the model creation. As with the other methods, it has weaknesses and is noted for its tendency to overfit data [12]. As with k-means, it requires user specification of the number of models k. Additionally, the use of a Gaussian distribution for fitting clusters makes a strong assumption about the distribution of the data in the feature space.

Self-organizing maps (SOMs) are an artificial neural network (ANN) approach to the clustering problem. It is a two step process to achieve clustering, requiring a training of the neural network model as well as a mapping step that maps the data to clusters. It constructs a set of neurons with connections that each represent a cluster and the relationships between the clusters. In this way, a topological map of the clusters and similarities is developed. Additionally, the analysis of weights in the network allows for the user to obtain a sense of which inputs are contributing heavily to which clusters. It also is understood to behave similarly to k-means for a small number of nodes [128]. As with k-means, the random initialization associated with the weights impact the results,

and the cluster assignments may vary based on the initial parameters. As with ANNs in other approaches, the rationale behind the setting of weights during optimization in the training of the algorithm can be difficult to ascertain, and the models contain a level of complexity that can be difficult for users to understand and interpret.

These clustering algorithms all require the identification of any appropriate transformation or standardization of the data, as well as the selection of a distance measure. The details of the implementation of a clustering algorithm must be considered within the framework of the dataset, and the unique properties of different training domains may impact cluster algorithm performance. A discussion of these issues given the considerations of different training domain datasets is presented in Chapter 3.

Another important use of unsupervised learning is for dimensionality reduction. As computer-based training technologies allow the collection of greater numbers of features, it can be important to trim the feature set prior to the use of supervised learning algorithms to prevent overfitting. The next section discusses dimensionality reduction and the common methods currently used in other domains.

Dimensionality Reduction

Dimensionality reduction (a form of feature selection) allows for the representation of data from high dimensional space in a lower-dimensional space. There are two primary reasons for this transformation: to remove redundant or irrelevant features, or to try to avoid the "curse of dimensionality." The first is fairly self-explanatory, where it allows the user to identify and remove features that are not providing useful content, which helps computation time and can prevent overfitting. The second relates to the idea that as the number of dimensions increase, the separation between points approaches a constant value. This creates great difficulties for algorithms that require measurements of distances or partitions of space, such as clustering algorithms and Support Vector Machines (SVMs, discussed later in the Supervised Learning section). A discussion of these considerations with regard to training domain data is discussed further in Chapter 3.

The main strategy for dimension reduction revolves around feature extraction, which involves the transformation of data from high dimensional space into lower dimensions. One set of methods that can be used are downselection methods, including forward selection and backward elimination. In forward selection, features are added to the model

iteratively, adding the feature that provides the best improvement in performance. Backward elimination operates similarly, starting with all features and sequentially removing the least useful feature each iteration. When a user-specified number of features are met (which must be fewer than the original number of features), the algorithm terminates. These methods can help identify which of the current set of features are the best for creating machine learning models but do not generate new features in themselves. Consider an example of trying to identify the 20 most important genes for a genetic illness out of a set of 1000 genes. Forward selection would find the best single gene for predicting the illness, followed by the second gene that best supports the prediction given the first gene, and so on until a set of 20 genes were identified. Backward elimination would begin with the full 1000 genes, and remove the gene that reduced prediction performance by the least (leaving 999 genes), and continue until only 20 genes remained. As can be seen, the best choice of algorithm may strongly depend upon the initial and desired number of features.

A different strategy comes in the form of Principal Component Analysis (PCA), which performs a linear mapping of the data into lower dimensional space [12]. The objective of PCA is to obtain a smaller set of orthogonal projections along the original feature space in a way that the variance of the data along the new dimensions is maximized. By using the first several components generated, dimensionality reduction can be achieved. A similar strategy is found in Linear Discriminant Analysis (LDA), which also looks for linear combinations of variables to express the data. However, LDA explicitly models the differences between classes (discrete labels) in data, while PCA does not require the specification of classes during the model construction [129]. Thus, LDA is intended for classification approaches (which use discrete labels) and not for regression approaches (which use continuous labels). PCA, on the other hand, is appropriate for both strategies. Since this thesis utilizes dimensionality reduction for input to both classification and regression, PCA was selected as the primary method for dimensionality reduction in the analyses presented in later chapters.

Unsupervised learning allows for a descriptive approach to the dataset, providing greater understanding of the structure of the data and the relationships between data points. For training domains, these algorithms could have uses in identifying trainees that are similar to one another (through clustering) or reducing the high number of

features provided by new computer-based training methods prior to supervised learning (through dimensionality reduction). Another important form of machine learning is supervised learning, which takes a predictive approach by constructing models that allow for the prediction of values on previously unseen data points. The following section provides background on supervised learning and some of the common algorithms used for supervised approaches.

2.2.2 Supervised Learning

Supervised algorithms infer a function that relates a set of input features to labels (also called "targets"). Supervised learning is directly related to prediction tasks, since once a model is learned the labels of new data points can be predicted based on their particular input measurements. For example, in a classroom environment, supervised learning could be used to identify the relationship between formative assessment measures such as quizzes to prediction targets such as the final course grade. For the analysis presented in later chapters, a set of commonly used supervised learning techniques were selected and are shown in Table 2.3, listed roughly in order of increasing complexity of the models constructed by the algorithm and covering a range of regression and classification approaches. These supervised algorithms can generally be divided based on the nature of the target variables; regression algorithms predict continuous target variables, while classification algorithms predict discrete target variables. Since training settings may include both discrete performance targets (such as a course grade) as well as continuous targets (such as power output from a power plant), both types of methods are considered here. A brief description of each algorithm is provided below.

Linear regression is one of the most common and simple algorithms in machine learning. Typically it uses a least squares criterion to find a set of coefficients β such that it minimizes the difference between the target variable y and the linear regression estimate $X\beta$. While simple, it makes several assumptions about the data. First, it assumes that the observations are i.i.d. (independent and identically distributed). Second, it assumes that the feature variables have no measurement errors. Third, it assumes that the target variable is a linear combination of the feature variables. Fourth, it assumes homoscedasticity, or constant variance in the errors of the target variable. Fifth, it assumes that the errors of the target variable are uncorrelated with each other. Sixth, it assumes that

Table 2.3: Details of Supervised Learning Techniques

Technique	Usage	Description
Linear Regression	Regression	Identifies linear parameters relating features to target continuous variable
Logistic Regression	Classification	Identifies parameters of logistic function relating features to target binary variable
Support Vector Machines (SVMs)	Classification	Identifies maximum-margin hyperplane separator, most commonly used in classification
Artificial Neural Network (ANN)	Regression / Classification	Creates predictive models using layers of neurons that have weighted connections and activation functions to produce outputs

there are no redundant features; that is, there is no multicollinearity in the feature space. Despite all of these assumptions, linear regression is a relatively robust method and has been found to be very useful in a wide range of domains.

Logistic regression follows a very similar strategy to that of linear regression, but instead feeds the feature variables through the logistic function. In this way, it produces a continuous output between 0 and 1. This can be useful in a binary classification approach, as this value can be interpreted as the probability that the data point is a member of class 1 (compared to class 0). By applying a cutoff probability (such as 0.5), we can provide a predicted classification for each data point. Logistic regression generally makes fewer assumptions about the target variable and assumptions seen in linear regression such as homoscedasticity are not required. It does, however, still maintain assumptions that the observations are i.i.d., feature variables have no measurement error, and no multicollinearity [130].

Support Vector Machines (SVMs) find the maximum-margin hyperplane that separates data in the feature space. Specifically, this can be expressed by the optimization problem shown in Equation 2.1. In this equation, $\cdot$ denotes the dot product, w is the normal vector to the hyperplane, and $\frac{b}{|w|}$ determines the offset of the hyperplane from the origin.

$$\begin{aligned} &\underset{w,b}{\operatorname{argmin}} \frac{1}{2}\|w\|^2 \\ &\text{subject to (for any } i = 1, \ldots, n\text{):} \\ &\qquad y_i(w \cdot x_i - b) \geq 1 \end{aligned} \tag{2.1}$$

Typically, SVMs have been used in a classification setting, but more recently have also been shown to be applicable to regression strategies. They are very flexible in the types of decision boundaries that can be created, being able to create both linear and non-linear boundaries through the use of kernels. Kernels allow for the finding of linear separators in higher dimensional space without the need to explicitly calculate the transformation of the data into the new feature space (for more information about kernel methods, see [12]). The assumptions behind SVMs are minimal, only assuming that the data are drawn independently and identically distributed (i.i.d.) from some unknown probability distribution [131].

Artificial Neural Networks (ANNs) are a class of algorithms inspired by biological nervous systems, involving layers of "neurons" that feed information through the network. The neurons contain adaptive weights that update based on a learning function. They do not make any *a priori* assumptions about the data [12], however they do require the user to specify the network structure which includes selecting the number of layers and the number of neurons within each layer. ANNs can be used in both classification and regression approaches, though they have the drawback that the complexity of the model makes it difficult for the user to understand the relationship between inputs and the resultant prediction.

2.3 Educational Data Mining (EDM)

In recent years, researchers have begun to apply machine learning algorithms to the closely related field of education. The use of machine learning in educational datasets has become a field in its own right and is often referred to as Educational Data Mining (EDM) (see [132–134] for a review of EDM literature). Much of the focus of this work has been on the increasing use of online education [135, 136], which generates enormous datasets that machine learning approaches are well-equipped to process. However, to date

there has not been an in-depth analysis of the nature of data generated in educational settings, and the impacts of data from these settings on machine learning applicability and results.

In traditional classroom settings, recording of student's progress is often a combination of subjective performance reviews and paper records [132]. The resultant data available for analysis are typically measures of cumulative knowledge (e.g. examinations), and often are only available at discrete points in the course progression. The increase in online learning settings has had several major impacts on educational data availability. Online and computer-based learning settings typically create an electronic log of a wide array of interactions between the student and the learning system, such as individual question responses (such as multiple choice or text-based) over an often more prolific set of homework, quizzes, and examinations. Data collection both in depth and quantity that would be impractical for traditional classroom settings is easily recorded through electronic venues. From a practical perspective, in these settings there is greater access to data at a finer temporal scale, since logs are typically recorded at every student interaction with the system. Additionally, the increased level of detail contained in the logs also often allows interpretation of process-level data. Rather than only being able to access the end result of each student's effort, intermediate pieces of information can provide insights into the progression of learning. It stands to reason that these qualities could improve the accuracy and usefulness of machine learning algorithms applied to these datasets, such as informing the timing and content of Trainee Interventions (TI).

There are several important applications of machine learning models identified by EDM literature. TI can be assisted by prediction algorithms through the identification of students that are struggling or are likely to have poor final course performance. In a targeted intervention approach, it is not only important to have high-accuracy predictions of students, but also to obtain these predictions as early as possible in the course, so that any interventions can be as impactful as possible in the student's understanding throughout the course. Another important application focuses not on the students, but rather on improving the course. The quality control task that identifies aspects of the course that are unclear or not meeting their learning objectives can also be informed by machine learning models. By using prediction models from a diagnostic perspective, those metrics that are the most valuable for student prediction (and therefore assessment)

can be identified. Additionally, unsupervised learning algorithms can identify patterns of errors across students, providing insight into those topics or assignments that pose the greatest difficulty for the students. It should be noted that while both supervised and unsupervised algorithms can provide insights into the course structure and the effectiveness of each of the course elements, they do not directly make recommendations for changes in course structure. Rather, the results must be interpreted by the educator to determine whether action is needed. These applications that have been identified in EDM literature have parallels in training, and help inform the potential applications to training presented in this thesis in Chapter 3.

2.4 Chapter Summary

The first section of this chapter provided a theoretical and practical background in training and training evaluation. Several training domains were discussed, and important considerations for training evaluation methods were identified. The second section of the chapter introduced common machine learning methods and their advantages and disadvantages. The third section presented applications and lessons drawn from EDM literature. With this background, Chapter 3 presents considerations in the application of machine learning algorithms to training data under several types of training domains. Chapters 4 and 5 then present the results of the application of machine learning approaches on representative datasets from knowledge-based and rule-based settings, respectively.

Chapter 3

Machine Learning Considerations for Training Applications

Chapter 2 presented background research in the areas of training and machine learning. A range of training domains and training task types were discussed, and machine learning algorithms and their properties were presented as well. This chapter focuses on the implications of this background information on the application of machine learning algorithms to training data. Specifically, what consequences the nature of training data (based on training domain) will have on machine learning algorithms and how the strengths and weaknesses of the various machine learning approaches manifest when applied to training datasets. This chapter is divided into four sections: the first provides a more detailed discussion of the typical data types provided by each training domain as classified by the SRK framework, the second presents two example scenarios from rule and knowledge-based training, the third discusses the use of machine learning algorithms on datasets from each of these environments, and the fourth presents a set of potential applications of machine learning approaches to training evaluation and trainee intervention (TI).

3.1 Data from Training Domains

As discussed in Chapter 2, data gathered as part of training evaluation may exhibit markedly different properties from those typically used by machine learning algorithms. Briefly, machine learning algorithms typically require a large amount of data, a high dimensional feature space, and consistent noise. Training data, on the other hand, may

not meet some or all of these conditions. This section expounds some of the typical properties of training evaluation datasets.

3.1.1 Number of Data Points

Restrictions on the number of available points (instances) in the data usually arises from a limited number of trainees completing the training program. Except for very large organizations, the number of trainees for a particular position may number in the tens to hundreds, well below the typical datasets used in machine learning approaches (which typically use thousands of data points). Low number of instances is a common issue experienced in human subject testing. Compared to the testing of machinery or computer programs, gathering data on humans typically requires much greater time and effort. In the case of training, the number of data points will correspond to the number of trainees completing the training program. While an organization could pay for additional people to complete a training program just to collect data, thereby increasing the number of data points available for analysis, usually this will be too expensive and an inefficient strategy. Changes in training program structure or content also present a problem for accumulating large numbers of data points. By their nature, machine learning algorithms rely on learning relationships in data based on a set of features. If the number or nature of the features change from one trainee to another, the data points cannot be combined to train the learning algorithm. Consider a case of trying to model the relationship between trainees' ratings of a course and their final score in the course. If the ratings were only gathered for half the trainees, only these trainees can be used to create the model relating these two variables. Since training programs may change frequently, either based on prior training evaluation or due to changing job requirements, gathering a sizeable dataset that is usable for machine learning may be challenging.

The particular type of training as defined by the SRK framework will impact the expected number of data points. Training for skill-based and rule-based domains are often based upon a high level of repetition. Thus, there will typically be greater opportunity for multiple trainees to complete the same task or the same trainee to complete a task multiple times than in knowledge-based domains. Additionally, skill- and rule-based domains have the highest need for retraining, providing additional opportunities for gathering data on the same training program from experienced personnel. Knowledge-based

training often focuses on learning abstract concepts or developing mental models to apply to a wide variety of scenarios. To accomplish this, training programs for these domains often include novel scenarios or problem solving. Data from training programs that utilize different assessment methods or questions (e.g. different scenarios and problems) with each set of trainees will not be as conducive to the use of machine learning approaches.

3.1.2 Number of Features

Features are the variables utilized as inputs to the machine learning models. Whether in a descriptive or predictive sense, the selection of features will determine the ability of the models to describe relationships in the data, as they provide the information used by the models. In a training program, the data is provided by the assessment metrics gathered, and the set of features must be selected from these metrics. Ideally, these will be features that are informative: they are able to tell trainees apart, have a relationship with the target variables (when used for prediction), and do not replicate information contained in other features.

The number of features in human training datasets will primarily depend upon the trainee assessment methods used during training and is thus also dependent upon the domain. As discussed in Chapter 2, skill- and rule-based training is traditionally dominated by subjective feedback, either self-assessment or assessment from an expert instructor. These subjective ratings will often have relatively few elements. For example, a study by the Health Education and Training Institute gathered only 19 ratings made by trainees [137]). Even when administered multiple times over the course of the training program, these summative assessment methods will rarely account for greater than tens to hundreds of features. Datasets from classic machine learning domains (e.g. genetics) will typically have thousands of features or more.

More recently, the use of simulators and CBT methods allow for the collection of many more interactions between the trainee and the system. Most commonly this will come in the form of log files that record time-stamped events in the system. These events will either be automatically generated by the simulation (such as an intentional error to challenge the trainee) or created through the interactions of the trainee with the system (such as clicking on an interface). As discussed in Chapter 2, these logs contain "process-level" information about how the trainee moves through each module

in the training program. This information provides a much greater number of features for use in machine learning approaches by allowing each individual action to be used as a feature, in addition to the traditional summative assessment metrics. Figure 3-1 depicts the differences in the specificity of information between traditional and computer-based training assessment approaches.

While process-level information from computer-based methods can provide a much higher dimensionality to the feature space, there are additional considerations in using this information to train machine learning algorithms. In machine learning, it is generally desirable to have a significantly higher number of points in the data than features in order to limit overfitting of the training dataset. The problem becomes apparent given an example: consider a training dataset with 20 trainees who have each completed 20 simulation runs (giving 20 data points and 20 features), and the evaluator wants to predict post training performance. The algorithm trained on this data can fit one feature

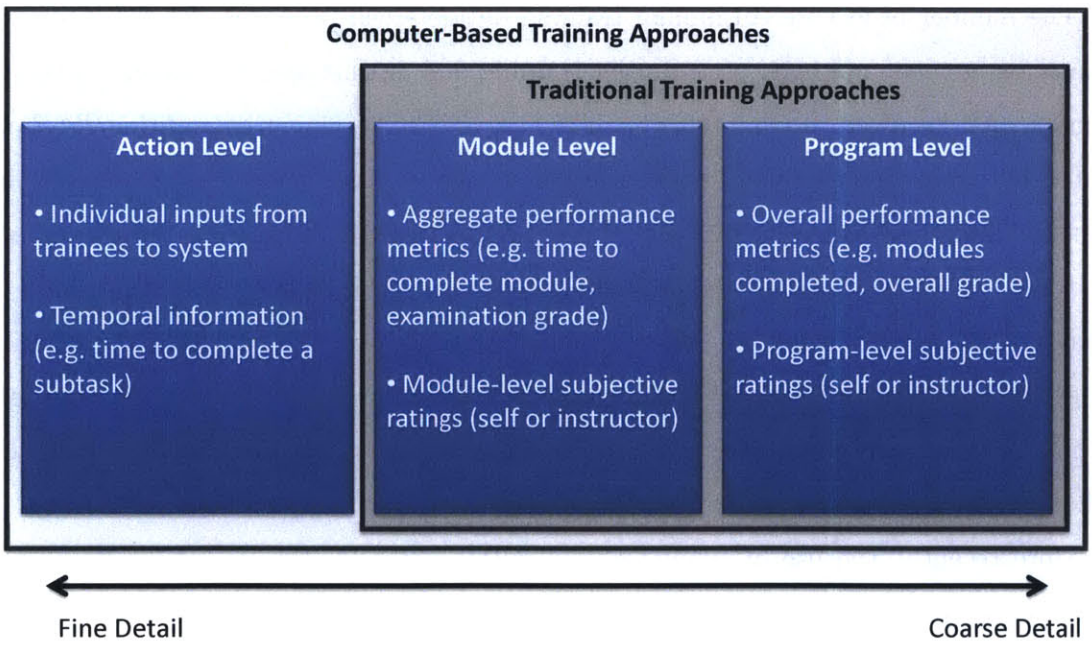


Figure 3-1: Comparison of specificity of training information available from traditional and computer-based approaches.

(simulation run) to each trainee (data point) to exactly predict everyone in the training dataset. In this case, the model has used all available features as part of the modeling, including modeling any noise in the data. While this model would perform perfectly on the training dataset, its generalization performance on a previously unseen test dataset would be very poor. The concept behind this issue is typically referred to as the "bias-variance tradeoff" [138]. Briefly, as the complexity of a model increases (i.e., number of features), the model tends to use all available degrees of freedom to train the algorithm. Any real-world data set contains both true information (signal) and variation (noise) within it, and these available degrees of freedom are used to fit the noise within the data set. Thus, while it is beneficial to have a high number of features, having too many features relative to the number of data points can also generate poor performance for a given model. It is very difficult to define an exact ideal ratio of features to data points, since it will depend upon the signal-to-noise ratio in the data. However, it is generally accepted that if given m features and n data points, it is desirable to have $n \gg m$.

Thus, while there may be useful information contained within the process-level features, additional adjustments have to be made in the application of a machine learning approach. Specifically, there are two main strategies for dealing with a low instances-to-features ratio. The first, feature selection methods, has already been discussed in Chapter 2. In this approach, features are either removed or combined to reduce the number of available features in the model. The second strategy, boostrapping, creates duplicate datasets from the original instances of data to help control for overfitting when using a larger number of features [139]. Bootstrapping makes the assumption that the original dataset is representative of the population, which may be a difficult assumption to meet in a small and high-noise dataset such as is seen in training data (the following section provides additional discussion of noise). Therefore, for the purposes of incorporating process-level information into machine learning approaches in this thesis, feature selection methods are utilized to reduce the number of features used and thus increase the instances-to-features ratio.

3.1.3 Noise

As mentioned in the previous section, the signal-to-noise ratio plays an important role in model performance. Noise in the data also plays other roles in machine learning

approaches, particularly with respect to the assumptions of machine learning algorithms. Some of the algorithms described in Chapter 2 assume homoscedasticity, or constant variance in the data. This assumption is very difficult to meet in human performance data, as the variances in performance data across trainees are very unlikely to be equal. Additionally, by the nature of training, performance data will not be consistent over *time*. As a trainee completes the program, it is expected that average performance will improve and variation in performance will decrease as the trainee becomes more familiar and more practiced with the system. Thus, the nature of noise in training data may not be conducive to machine learning methods that have hard assumptions about data variance.

Another aspect of noise in training data that is unusual in machine learning approaches is the change in variance across features. Assuming the performance data can be fit to a distribution, some trainee assessment metrics may follow a normal or nearly normal distribution (such as the attitude error in a flight simulator). However, other metrics might follow other patterns like a lognormal distribution (such as time to notice a deviation in a monitoring task). While machine learning approaches can be run using features that are generated under fundamentally different processes, this violates assumptions of most algorithms about the nature of noise in the dataset. One strategy to help with this issue would be to transform or normalize the data (discussed later in this chapter), although implementing a transformation that is fundamentally different than the underlying distribution does not guarantee the success of a machine learning approach.

While issues relating to number of data points, number of features, and noise may occur in data sets generated from any training environment, the severity will depend upon the specific domain and assessment methods. To illustrate these issues and provide specific examples of typical assessment techniques and the resultant data, the next section presents two example scenarios from rule- and knowledge-based training domains.

3.2 Example Training Datasets

As discussed above and in Chapter 2, the increase in the use of simulators and CBT as part of training programs may improve the suitability of machine learning approaches

on training datasets. These elements have been primarily implemented in rule-based and knowledge-based training environments, such as training for pilots, nuclear reactor operators, Air Traffic Control (ATC) operators, or students through Massive Open Online Courses (MOOCs). There have also been advancements in computer technology and simulation in skill-based environments, such as instrumentation of sports equipment or optical tracking data of player position (e.g. [140]), but these systems are typically recently deployed and use proprietary data. While machine learning approaches may also be applicable to datasets collected by these systems as they mature, this thesis focuses on particular cases from the more established technologies in rule-based and knowledge-based training.

To illustrate the typical training dataset properties described above, two examples are provided here. The first focuses on simulator training for commercial pilots, generally a rule-based environment. The second is a principles of flight course, which focuses more on knowledge-based training. These two training areas are fundamentally related in topic (aircraft) but take markedly different approaches to training methodology. As a result, the typical training data available from each of these sources will exhibit different properties, potentially impacting the usefulness of machine learning approaches.

3.2.1 Pilot Training

For pilots in commercial aviation, a significant amount of training is required before a pilot is allowed to fly an aircraft, and even experienced pilots must undergo frequent refresher training. There are a wide range of training programs for pilots mandated by aviation authorities such as the US Federal Aviation Administration (FAA) and the UK Civil Aviation Authority (CAA). This example will focus on a particular part of commercial pilot training, known as Line-Oriented Flight Training (LOFT). LOFT makes use of high-fidelity flight simulators to recreate scenarios that may be experienced during operational flying. Some basic elements of LOFT are listed below [141]:

- LOFT should take place in a line operational environment with a complete crew
- LOFT should contain scenarios of real-world, line operational situations, which progress in real time

- The simulation is allowed to continue without interruption, even when mistakes are made

The FAA specifies four phases of LOFT: briefing, preflight planning and activities, flight segment, and debriefing [141]. In the briefing phase the instructor informs the crew of the training objectives, environmental settings, role of the flight crew, and role of the instructor. During the preflight planning phase, documents such as weather reports and baggage weight are provided to the crew, allowing them to prepare the appropriate flight plan. The flight segment includes taxi, takeoff, flying, and landing as well as communication with Air Traffic Control (ATC). Debriefing provides feedback to the crew on their performance from the instructor both to individuals and the team as a whole.

Trainee assessment in LOFT is primarily done by the instructor and is provided as feedback during the debriefing phase. To prevent employees from modifying their behavior due to concerns about losing their employment, it is understood that the pilot or crew will not be disqualified even if a serious error is made during the simulation (e.g. crashing the airplane) [142]. Rather, LOFT is intended primarily as a learning exercise, and even with serious errors a "satisfactory completion" rating is achievable, so long as the mistakes were obvious to the trainee and were judged to not need further attention. The instructor will inform the crew of any concerns during debriefing, but it is extremely rare for a crew to "fail" a LOFT scenario. This subjective feedback given by the instructor comprises the majority of trainee assessment during LOFT. The simulator is also capable of recording the states of the system in log files, and objective performance could be obtained for the purposes of machine learning approaches; however, this data is not currently used for assessment.

For this example a Pan Am LOFT scenario of an A-310 flight from Washington-Dulles International Airport (airport code IAD) to John F. Kennedy International Airport (airport code JFK) is selected [142]; the details of the scenario can be found in Appendix A. This scenario consists of thirty-four elements, including two problems built into the scenario. These thirty-four elements can be grouped into a set of main stages. First, the simulation is set up with all the appropriate parameters for the scenario. Next, the crew completes their preflight checklist, while communicating with ATC and ground crew to receive clearance for starting the engines and pressurizing the hydraulic systems. During this stage the crew encounters the first problem, an engine hot start (improper combus-

tion). After addressing this problem and completing the preflight checklist, the crew is given clearance to push back from the gate and taxi to the runway. After arriving at the runway, the crew requests clearance to take off and completes the takeoff checklist. After takeoff, the crew follows the planned flight path and ATC guidance to JFK airport. While nearing JFK airport, the crew encounter their second problem, either an emergency landing at an alternate airport, a passenger threat, or a communication failure. Depending upon the issue, the crew resolves it and lands the aircraft while completing the approach checklist and communicating with ATC. The scenario ends after arriving at the gate, communicating with ground control, and completing the arrival checklist.

Based on this scenario layout, there will be several important aspects of the scenario that can be used for assessment, beyond the current assessment methods that only use instructor feedback. It is apparent that the pilots use a multitude of checklists during their normal flying routine. The completion of these checklists, which represent rule-based training, can be analyzed in several ways to generate features for a machine learning approach. The number of checklist items that were skipped or transposed could be counted, as well as actions that are added during checklist completion that were not intended to be performed. These actions could include modifications to the system (such as turning on a hydraulic pump), communications to ATC or ground control, or internal communications and checks for the crew. These are all measures of the accuracy to which the checklists are followed and are termed "procedure adherence" measures. In addition to procedure adherence measures, temporal metrics can be examined, such as the completion time for a particular action or checklist. In a rule-based training environment, these procedure adherence and temporal metrics are important features to consider for machine learning approaches. A further development of the analysis of these features is described later in this chapter.

Another aspect that could be utilized for machine learning are the flight performance characteristics during the scenario. These could include the heading, airspeed, attitude, descent/ascent rate, GPS position, flaps settings, or even the individual yoke movements made by the pilots. Deviations from the expected values of these parameters can be recorded and used as assessment metrics. Since the recording of these values can be done relatively frequently (>1Hz), they can provide high-density performance information. As features in a machine learning approach, however, it would be illogical to include

all recorded events as separate features, due to issues with overfitting. Instead, these performance metrics could be condensed through feature selection techniques or could be aggregated over more meaningful periods of time, such as the average error in heading during the flight segment of the scenario.

This setting provides an example scenario for a primarily rule-based training environment and the assessment metrics that would commonly be generated in such an environment. The use of machine learning methods on data from this training example will be further discussed later in this chapter . The next section presents the other example scenario, a knowledge-based training environment from a principles of flight course.

3.2.2 Principles of Flight Course

The example case selected for the knowledge-based training environment is a course in Principles of Flight from the Joint Aviation Authorities (JAA) [120]. This course covers a wide range of theoretical knowledge related to flight. The main topics of the course are: subsonic aerodynamics, high speed aerodynamics, stability, control, limitations, propellers, and flight mechanics. Within each of these topics are many subtopics relating to concepts such as stall or drag, each with their own learning objectives. While these are too numerous to list here, an example section on drag and lift is shown in Table 3.1.

From the learning objectives provided by this course section, it is apparent the goals of the program relate to the theoretical and conceptual understanding of the trainees. Assessment techniques will therefore aim to assess the understanding of the trainees as they move through the course. As discussed in Chapter 2, summative measures are traditionally used to assess the comprehension of the trainee and the ability of him/her to extrapolate the knowledge into new scenarios. This could take the form of quizzes, projects, presentations, or examinations that require the trainee to utilize the knowledge they have gained. Typically, the specific questions used in assessment will not have previously been seen by the trainees (or at least not heavily practiced), requiring the trainee to draw upon long term memory and cognitive models to answer. For written assessments, common formats include multiple choice, short answer, or essays. For oral assessments or passive assessment (without direct input from the trainees), subjective ratings by the instructor remain the most common. Subjectivity still plays a fundamental role in written assessments as well (except for multiple choice); the instructor must judge

how well the trainee answered the question or demonstrates the desired understanding. Usually the assessment scores are placed upon a numeric scale, such as 0-10 or 0-100 (often representing a percentile). Compared to rule-based environments, scores across trainees may tend to better approximate a normal distribution. However, summative measures by definition intend to broadly assess topics and are usually only captured at certain distinct points in the course. Thus, in a typical knowledge-based course the number of possible features to use in a machine learning approach may be limited.

More recent developments in knowledge-based training such as MOOCs typically provide assessment after short presentations of material [143], allowing for the collection of much finer detail of the learning process of the trainee. In these methods assessment would be taken during and after each topic and subtopic discussed above (e.g. identify the significant points on the lift and angle of attack graph), providing a large number of features for use in machine learning approaches. If they contain useful information about the learning of the trainee, these "process-level" features could contribute to the accuracy and usefulness of machine learning models in these datasets. The implementation of machine learning algorithms on these datasets is discussed further in the next section of this chapter, while the results of the applications of these methods on an example dataset is presented in Chapter 4.

Table 3.1: Sample Learning Objectives from Priciples of Flight Course

Topic	**Learning Objectives**
Drag and wake	List two physical phenomena that cause drag. Describe skin friction drag. Describe pressure (form) drag. Explain why drag and wake cause a loss of energy (momentum)
Influence of angle of attack	Explain the influence of angle of attack on lift.
The lift - α graph	Describe the lift and angle of attack graph. Explain the significant points on the graph. Describe lift against graph for a symmetrical aerofoil.

3.3 Machine Learning on Training Data

The previous section introduced example datasets from the rule-based and knowledge-based training domains and discussed the typical properties and assessment measures

available for machine learning approaches. This section discusses some of the considerations of applying machine learning approaches given these properties. The first subsection discusses selecting features from each dataset, while the second provides a discussion of the other preparations needed when applying machine learning approaches to each example dataset.

3.3.1 Features

Rule-based training focuses on repetition and is often conducted using simulation environments. Logs of interactions between the trainees and the training system allow for a variety of possible machine learning features and techniques to be used. This section outlines some of the details of selecting features from these datasets and other considerations in the application of machine learning approaches.

Rule-based Training

As previously discussed, the two main data types that arise in rule-based domains such as the pilot training example involve procedure adherence and simulation performance metrics. In a procedural setting, there is a specific order to the actions to be taken, as indicated by the procedures. Procedure adherence metrics can be collected by action (i.e. "was this action the correct one?") or at the end of a procedure or module, such as total number of skipped actions. At the action level, the simplest assessment metric would be a binary of whether the action performed was correct or incorrect. However, if the action is incorrect, this basic metric does not provide a gradation of *how* wrong the action taken was. Consider a case where the trainee is supposed to check the hydraulic pressure and then check the pump status. If the trainee first checks the pump status and then the hydraulic pressure, one could argue that, while incorrect, these errors might not be as serious as if they had incorrectly shut off a pump. It is not easy to define a scale for the "incorrectness" of an action, but two possible strategies include considering how similar the action was to the correct action and the safety implications of executing the incorrect action. In either case, this provides an alternative metric for each action, indicating not just whether the action was correct or incorrect, but a gradation of the correctness of the action. In Chapter 4, which analyzes data from a rule-based environment, both of these strategies are considered. There are also summative procedure adherence metrics that

are available at the end of each module or at the end of the program. These could include agglomerative versions of the prior metrics discussed, such as total or average error from the action-level.

Another set of assessment features that could be included are the performance metrics of the trainee in a simulation environment. In the pilot training example, this could include metrics such as heading deviation or descent rate at landing. These will be specific to the domain and the simulation environment used in the training, and it is difficult to suggest general performance metrics for use as features for machine learning. However, a few overall observations can be made about performance data in the context of machine learning. First, the objective nature of the data make them well suited for machine learning approaches, as there may be more consistency in the measurement across trainees. Second, since the simulator itself has control of the training environment, these metrics are unlikely to have large measurement error. Many machine learning approaches assume no measurement error, and performance data may fit that assumption. Third, in a real time simulation environment such as the pilot training described earlier, the frequency of recording performance measures may be quite high. This creates both benefits and difficulties for machine learning approaches. High frequency recordings allow for a high number of features, which is generally beneficial to machine learning approaches. However, if the number of trainees (data points) is limited as in many training domains, too many features can create issues of overfitting as discussed earlier in this chapter. Therefore, it is appropriate to determine summative performance measures or utilize feature selection techniques as described in Chapter 2 to limit the size of the feature space while still making use of the information contained in the peformance data.

Traditionally the most common assessment technique in cases such as pilot training relate to instructor subjective ratings of trainees. These could be utilized as features in a machine learning approach but have been shown to have difficulties in consistency across trainees [110–113]. In particular, if there is error in the measurement of these metrics (i.e. variations in the judgment of the rater between or within trainees), these features may not be as useful in a machine learning approach. However, these features certainly maintain their usefulness *in addition to* machine learning in training evaluation. Maintaining these rating systems would not only improve the consistency and familiarity of the instructors with the trainee assessment process, but they may still be used in the decision making

process of the instructor about the progress of the trainee and any possible need for TI. A further discussion of the relationships between traditiional metrics such as ratings and machine learning outputs in the context of instructor decision making is presented in the last section of this chapter.

Knowledge-based Training

In knowledge-based training environments, the most common data type available is a scaled score, usually on a scale of 0-100. For each assessment (quiz, test, project, presentation), the trainee is assigned a scaled score. While these scores can be objective when using assessment methods such as multiple choice questions, often these are subjectively evaluated by the instructor. Thus, many of the features inherent to knowledge-based environments will have the same issues with consistency and validity associated with subjective rating systems, as discussed in Chapter 2. As presented in the previous rule-based section, subjective data may result in reduced machine learning performance due to the presence of measurement noise. But since these subjective metrics remain the dominant assessment technique in knowledge-based domains, they will play an important role in machine learning on these datasets. Thus, any machine learning approach in a knowledge-based training environment must address the limitations associated with this type of data.

An additional consideration that arises in knowledge-based domains is that usually the overall assessment of the trainee will be based upon a combination of unequally-weighted individual measurements. Specifically, summative measures such as examinations usually have more weight in determining the overall performance of the trainee than quizzes on individual topics. In a machine learning approach, the unequal importance of these features must also be incorporated to avoid skewing the results. While there may be cases where it is desired to treat all features equally, in most knowledge-based domains the assessment methods are designed to have differential importance in assessing learning. From a machine learning perspective this is particularly important in cluster analysis, which is discussed further in the section on distance measures below.

3.3.2 Other Considerations

Once features have been identified and the algorithms have been selected (both unsupervised and supervised algorithms identified in Chapter 2), there remain several additional considerations that must be addressed prior to the creation of the machine learning models. These are data standardization, selecting a distance measure for unsupervised approaches, and selecting metrics for the comparison of performance across models.

Data Standardization

For any machine learning method, the relative scaling along any particular feature may dramatically impact the weights associated with those features. One possibility to reduce the influence on arbitrary scaling is to standardize or transform the data. This approach is particularly important in unsupervised clustering methods. Any clustering approach requires that the algorithm compute distances in the feature space. Without weighting, it would be inappropriate to directly calculate a distance measure (such as Euclidean distance) where one feature has a range of 0-1 while another has a range of 0-10000. There are two main options to address the potential lack of comparability of features: feature weighting (which has already been mentioned) and standardization. Standardization involves the normalization of all features to a common scale, such that distances along each feature carry similar meaning. The most common standardization strategies include range transformation and z-score transformation. The range transformation is given in equation 3.1

$$range(x_{if}) = \frac{x_{if} - min(f)}{max(f) - min(f)} \tag{3.1}$$

where x_{if} is the f^{th} feature value for the i^{th} data point, $min(f)$ is the minimum value of the feature in the dataset, and $max(f)$ is the maximum value.

The equivalent z-score transformation formula is given in equation 3.2

$$z(x_{if}) = \frac{x_{if} - m_f}{s_f} \tag{3.2}$$

where x_{if} is the f^{th} feature value for the i^{th} data point, m_f is the mean value of the feature across all data points, and s_f is the sample standard deviation of the feature across all data points.

In range standardization, all data points are mapped to a value between 0 and 1 on all features, with 1 being the maximum and 0 being the minimum. In z-score standardization, the feature is fit to a standard normal distribution, and transformed values for the data points on this distribution are calculated. The z-score transformation generally operates well when the feature values tend to naturally fit to a normal distribution. For unusual distributions of values over the feature, range standardization may be preferred.

In a rule-based environment such as the pilot training described above, transformations on procedure adherence data are only appropriate for summative metrics or action-level metrics that are graded by "correctness" as described earlier. For both of these types, the selection of standardization strategy will depend upon the relative normality of the distribution over trainees. In general, it would be expected that since all trainees are striving for perfect adherence, it is unlikely that the distribution of actual adherence metrics would generally fit a normal distribution (i.e. an "average" error plus or minus some deviations). The intent of these environments is not to confuse the trainees and induce errors, and as the trainees gain practice in these environments their performance will tend towards perfect adherence. Therefore, it is generally anticipated that a range transformation would be most appropriate for adherence data. Alternatively, performance data such as mean-squared-error (MSE) have been shown in flight simulators to have much better approximations to normal distributions across trainees. Thus, it is worth considering z-score transformations on performance type data from rule-based training.

In knowledge-based training environments, much of the data will be provided on a scale of 0-10 or 0-100. For most of the features that fall on these scales, a z-score transformation is appropriate and is commonly used in education for providing statistics about assessments (e.g. average and standard deviation of scores on an examination). However, as will be discussed in Chapter 4, the use of frequent low-value assessments in online courses may create difficulties in using a z-score transformation. Specifically, if some trainees do not complete an assessment (and therefore receive a score of 0 for that metric), this clearly is generated under a different distribution than the scores of those who completed the assessment. In these cases, a z-score transformation is not appropriate and a range transformation should be used. A general heuristic that can be used is considering whether all data for the feature was generated under a consistent process. If so, a z-score

transformation may be appropriate; otherwise a range transformation will be more robust to unusual distributions across trainees.

While these general rules may be considered for rule-based and knowledge-based domains, an empirical investigation can be conducted to determine which transformation performs better in machine learning approaches. Empirical results for two example cases can be found in Chapters 4 and 5.

Distance Measures

The selection of an appropriate distance measure is important for unsupervised learning approaches. There are a wide range of distance measures that have been proposed for use in clustering algorithms, and in fact virtually any norm function that assigns a positive length to each vector in a vector space can be used as a distance measure. A selection of the most common measures are presented in Table 3.2.

Generally, for a given set of differences between two trainees, Euclidean distance will penalize more heavily when the differences are on few features, while the Manhattan distance will provide the same penalty whether the deviations are across many features or only a few. To better illustrate this difference, consider a case from the Principles of Flight example above, where two trainees (Bob and Jenny) have taken two quizzes, one on lift and one on drag. If Bob scores 80 out of 100 on both quizzes, while Jenny scores 90 out of 100 on both quizzes, the Manhattan distance would simply sum the differences and rate the two trainees as having a distance of 20. The Euclidean distance, however, would calculate the square root of squares of individual feature distances would result in

Table 3.2: List of Common Distance Measures

Distance Measure	**Formula (assuming d features)**	**Usage**
Euclidean Distance	$dist(x_i, x_j) = \sqrt{(x_{i1} - x_{j1})^2 + \ldots + (x_{id} - x_{jd})^2}$	Most common distance measure
Manhattan Distance	$dist(x_i, x_j) = \lvert x_{i1} - x_{j1} \rvert + \ldots + \lvert x_{id} - x_{jd} \rvert$	Common distance measure, also called "city block" distance
Chebychev Distance	$dist(x_i, x_j) = max(\lvert x_{i1} - x_{j1} \rvert + \ldots + \lvert x_{id} - x_{jd} \rvert)$	Equivalent of the number of moves it would take a king in chess to move from x_i to x_j

a distance of $\sqrt{10^2 + 10^2} = \sqrt{200} \approx 14.14$. Thus, by having the 20 points spread over two features, the Euclidean distance treats the two trainees as being more similar than the Manhattan distance. Calculating the Chebychev distance results in simply finding the greatest difference in score across all assessments (in the example given, this would simply be 10). However, the Chebychev distance may have difficulties in cases where not all trainees complete every metric. A trainee who skips a single assessment (and is assigned a score of 0) would immediately have a large distance from all trainees who did not skip the same assessment. It is undesirable to have the distance metric become dominated by individual cases, particularly when process-level data is included in the analysis.

The difference between Manhattan and Euclidean distance increases with increasing numbers of features. The distance as measured by Manhattan distance will increase linearly: increasing from 10 to 20 dimensions will double the distance. This has the advantage of maintaining a clear meaning to the observer: it directly translates into the sum of differences in scores across the students in question. Regardless of feature space dimensionality, an increase on a single score by one student of 1 point compared to the other student will result in an increase in Manhattan distance by 1. However, this may not always be beneficial at high numbers of features. In both rule-based and knowledge-based training, inclusion of process-level information will dramatically increase the dimensionality of the feature space. In these spaces, Manhattan distance will become dominated by the process-level features. Euclidean distance will be impacted, but as the dimensions increase the impact of any particular feature will be lessened. Consider our example of the Bob and Jenny who tend to score 80 and 90, respectively. At 1, 2, 3, and 4 features, the Manhattan distance will be 10, 20, 30, and 40 as described. The corresponding Euclidean distance will be 10, 14.14, 17.32, and 20. Thus the impact of each added feature on the distance measure is depressed, which may be desirable if each individual process-level metric may not be as important as summative metrics.

Another strategy for addressing the differences in relative frequency and importance between process-level metrics and summative metrics can be handled by weighting. Providing weights to features in the calculation of distance can account for qualitative differences between the features. This strategy particularly makes sense in many knowledge-based environments, where the final performance in the course may be determined based

on a cumulative scoring across many assessment metrics. However, typically some assessments (such as examinations) make up a greater proportion of the final score than others (such as quizzes). Weighting features can specifically input these differences into the machine learning approach. Chapters 4 and 5 provide additional empirical results behind the selection of distance measures and the use of weighing of features.

Metrics for Comparison

The metrics used to compare machine learning approaches will differ fundamentally based on the type of algorithm, particularly between unsupervised and supervised algorithms. In supervised algorithms, the prediction of labels is the primary objective and prediction accuracy is almost always used as a performance metric for comparing machine learning approaches. Prediction accuracy takes on different forms dependent upon whether the labels are continuous (regression) or discrete (classification). In regression, the most common metrics give information on the "goodness of fit" of the model on the data, typically given as the coefficient of determination (R^2). For the analyses in later chapters, R^2 on the test dataset is used as the primary metric for comparison of model performance. The formula for R^2 is given in Equation 3.3.

$$R^2 = 1 - \frac{\sum(y_i - f_i)^2}{\sum(y_i - \bar{y})^2} \tag{3.3}$$

where y_i is each data point, f_i is the model prediction for that data point, and $\bar{y}$ is the mean of the observed data.

In classification, the most common metric for prediction performance is classification error. For each data instance in the test dataset, the model prediction of the class label is compared to the actual label, resulting in a binary "correct" or "incorrect". The percentage correct is computed, and this is reported as the classification error rate. For the analyses in later chapters, the classification error rate is used as the metric for comparisons of supervised classification model performance.

In unsupervised learning, defining metrics to compare models is more difficult. Since there is no "ground truth" provided by the labels in supervised learning, other strategies for model evaluation and comparison must be used. Generally, these can be divided into two categories: external, which are reliant on exterior data sources for comparison, and internal, which analyze the properties of the clusters created. Specifically, the external

measures of entropy and purity and the internal measures of compactness and isolation represent a set of common metrics for comparison [144]. For the purposes of the analyses in Chapters 4 and 5, all of these metrics are analyzed. In addition, it is widely recommended for the modeler to inspect the resultant cluster formations and provide subjective judgment on the performance of the algorithms as well. These judgments are included in the analysis in later chapters as well. A brief description of the external and internal measures utilized is provided next.

Entropy is an external metric for measuring cluster algorithm performance, which means that it makes use of the knowledge of "true" clusters. When labeled classification data is available (as in a supervised approach), the membership to the clusters found in unsupervised learning can be compared to the classes of the labeled points as a reference. In this sense, the classes are assumed to represent the "correct" cluster assignments. The entropy generally measures the randomness of the distribution of the classes amongst the clusters. Low entropy indicates that the distribution of classes among the clusters is fairly specific and thus indicates good cluster algorithm performance. The entropy associated with each cluster can be found as shown in equation 3.4.

$$entropy(D_i) = -\sum_{j=1}^{k} Pr_i(c_j) log_2 Pr_i(c_j) \tag{3.4}$$

where D_i are the points in cluster i, c_j are the points in the j^{th} class (from classification), and $Pr_i(c_j)$ is the proportion of class c_j in cluster i. In this formula, the minimum possible entropy (zero) is only achievable if all members of each class are placed only into a single unique cluster. If there are more clusters than classes, this can only occur if some of the clusters are empty. Thus, entropy is most appropriate in cases where the number of classes and clusters are the same. The entropy for each individual cluster can then be combined to find a total measure of entropy using the formula in equation 3.5.

$$entropy_{total}(D) = \sum_{i=1}^{k} \frac{|D_i|}{|D|} entropy(D_i) \tag{3.5}$$

As another external metric, purity also compares the clusters to the true classes in the data. Purity measures the extent to which each cluster contains only one class of data. High purity indicates good cluster algorithm performance. The purity of each measure can be found by the formula shown in equation 3.6.

$$purity(D_i) = \max_j(Pr_i(c_j)) \tag{3.6}$$

The total purity across all clusters can be calculated as shown in equation 3.7.

$$purity_{total}(D) = \sum_{i=1}^{k} \frac{|D_i|}{|D|} entropy(D_i) \tag{3.7}$$

Compactness and isolation are internal measures of cluster algorithm performance in that they do not require the use of additional labels. Compactness measures how near the data points in each cluster are to the cluster centroid, with algorithms that result in greater compactness being preferable. Compactness is usually calculated by sum-squared-error (SSE) of the data points within a cluster to the corresponding cluster centroid. This is formally expressed in equation 3.8.

$$compactness(D) = \sum_{i=1}^{k} \sum_{j \in \mu_i} (x_j - \mu_i)^2 \tag{3.8}$$

where k is the number of clusters, and μ_i is the centroid of cluster i.

Isolation measures the separation between the cluster centroids, as clusters that are far apart are desirable. Therefore, greater isolation is preferable in a clustering setting. This is typically found simply by the sum-squared-distance between the cluster centers. A formal definition is shown in equation 3.9.

$$isolation(D) = \sum_{i=1}^{k} \sum_{j \neq i} (\mu_i - \mu_j)^2 \tag{3.9}$$

where μ_i and μ_j are the cluster centroids of clusters i and j, respectively.

Expert judgments are also useful in analyzing clustering approaches, as the user may be able to identify particular characteristics or behaviors of the algorithms that tend to be desirable, such as strong clustering performance on an important group. All of these metrics are used in the comparison of the clustering algorithms presented in Chapter 2.

3.3.3 Summary

This section has discussed additional considerations for the application of machine learning algorithms to both rule-based and knowledge-based training data. Options for the

standardization of data, distance measures, and metrics for model comparison were presented. Empirical results from the use of these methods are presented for example knowledge-based and rule-based training data in Chapters 4 and 5, respectively. However, before the algorithms are tested on these datasets, it is useful to establish hypotheses for the potential applications of machine learning algorithms to training evaluation. Then, the performance of the algorithms can be compared against the desired applications. The remainder of this chapter presents the process for developing a set of potential applications that may be tested using the example training data presented in Chapters 4 and 5.

3.4 Applications of Machine Learning in Training Evaluation

As previously discussed in Chapter 2, there are a variety of improvements desired by the human training domain as discussed in the literature. One of the clearest is closing the gap between in-training assessment and post-training performance [15, 65]. More generally, the set of training needs can be divided into two categories: improvements to individual trainee evaluation and improvements to training program evaluation. The first is relevant to ensuring that each trainee that completes the program meets the performance requirements desired. The second focuses on feedback and improvement of the training program for all current and future trainees.

3.4.1 Individual Trainee Assessment

Individual trainee evaluation is a critical element of any training program and improved methods for evaluation are desired [28, 32]. These could take the form of novel metrics or improved analysis methods for metrics already used. Novel metrics include both those metrics that are not currently gathered by the training program as well as those that are gathered but not currently utilized for evaluation.

An ideal training evaluation metric will have several properties:

- Predictive - the metric relates well to post-training performance

- Discriminative - the metric can separate trainees from each other, thereby allowing trainees to be ranked according to performance
- Consistent - the metric has limited noise associated with the readings
- Convenient - the metric requires little effort to collect and analyze

A metric that meets all of these properties will be able to accurately and consistently assess trainees, and allow for the identification of trainees who will be successful in the field. These properties can be used to evaluate current metrics, and an example is presented below.

Consider training for a knowledge-based task such as the Principles of Flight example presented earlier in this chapter. Written examinations are one of the common assessment metrics for this environment. These types of assessments are highly discriminative and often provide a wide range of possible scores (often out of 100). These also represent cumulative knowledge accrued and aim to measure the capacity to which the trainee can apply the course material later in their educational and professional career [145]. However, written examinations are not as strong as evaluation techniques for the other properties. A particular student's performance may vary widely dependent on the specific topics or questions addressed in the examination [80]. While good trainees should generally perform better than poorer trainees, scores on any particular exam may fluctuate and this variation may have a strong impact on the interpretation of the trainee's learning. There may even be cases where not every trainee is evaluated by the same supervisor, or not every question within the examination is graded by the same person. Subjectivity of current evaluations is a widely reported issue (as discussed in Chapter 2) and it is apparent that new objective metrics that meet the desired properties would be useful to modern trainee assessment.

3.4.2 Assessing the Predictive Capability of an Evaluation Metric

As previously mentioned, an ideal training metric will be predictive of future performance. That is, by measuring data during training, one can get an idea of how the trainee will perform later on. Since training programs across all domains tend to be increasing in

complexity or difficulty with time (basic concepts are learned first), it seems logical that understanding a trainees mastery of the early material might give insight into how they will perform in the future.

There are several time horizons of interest for the consideration of predictive capabilities. First is post-training performance. This is the most desirable prediction to have, as it directly relates to efficiency, safety, and productivity in the field. However, it is also the hardest to assess, as the availability of field data may be far less than training data due to their cost and difficulty to collect. Predictions are of course tied to the data used, and performance predictions made in subsequent years from training completion become increasingly unreliable. If useful field data is available, there may also be administrative hurdles to using it in trainee evaluation. Such data collection would require significant use of on-the-job employee monitoring, and there are ethical and social implications that will no doubt arise.

The second time horizon of interest is predicting the final training program performance of a trainee. This still has high value for training programs if performance early in a training program can act as a predictor of final performance. If a trainee is having difficulties with material near the beginning of the training program, early recognition of this issue can allow the problem to be addressed in a timely fashion through TI. Since many training programs are typically long and costly, early intervention can result in savings of time or resources. Several possible interventions may be appropriate dependent on the case, including: attrition, where the trainee is removed from the program, or retraining, where the trainee is returned to earlier concepts to reinforce understanding. In either intervention case, the earlier the trainee is identified, the more savings can be obtained by the training program.

3.4.3 Improvements to Program Evaluation

The evaluation of the training program itself is also a critical element of training evaluation. Under the training paradigm, an investment of money and time is made in the trainee, in the hopes of obtaining benefits in the future. Ideally these benefits include efficiency, safety, or productivity. To justify the investment, there must be some method of assessing whether the benefits have been achieved across all trainees. This falls under the domain of training program evaluation.

At the highest level, training evaluation focuses on results: the changes in knowledge, behaviors, and ultimately the impacts to the organization. At a lower level, the elements that make up a training program can be dissected and analyzed. Typically, a training program is comprised of a series of training sessions, which herein are referred to as "modules". Each module usually covers a different topic and assists in the development of the overall skills and knowledge desired through training. One possible path to training program evaluation is to evaluate at the module level: is each module achieving its training objectives, and are these objectives useful in the completion of the entire program? If we continue to dig further, we could consider evaluating the usefulness of each data metric collected during each one of the modules. Another aspect of a training program that can be analyzed for improvement is training intervention. Both the timing and the content of interventions must be carefully planned to maximize their effectiveness.

While there are a variety of needs in the evaluation of training programs, it is not clear whether the capabilities of machine learning algorithms can meet these needs. The next section discusses the principal capabilities of different machine learning approaches, so that a set of potential applications to meet the needs of training evaluation can be derived.

3.4.4 Machine Learning Capabilities

There are numerous capabilities of machine learning algorithms, and some of their general characteristics have already been described in Chapter 2. Some additional specifics for each type of algorithm are detailed here.

Unsupervised learning algorithms generally fall under clustering and feature selection algorithms. In short, clustering algorithms ask: "given a metric, what are the similarities between data points?" while feature selection algorithms ask: "what metric should be used for judging similarity?" In clustering, the user provides a distance metric and all data points are compared and grouped using this distance metric. In its most direct application, these algorithms can be used to determine where natural groupings exist within the data; revealing these patterns and groupings to a training supervisor could have utility in a human training domain. Additionally, one could consider the inverse meaning to the clustering algorithms: rather than identify data points that are close together, we can identify those that are most unlike the others. In this way, we can

find outliers easily in high-dimensional space. Feature selection algorithms identify those features that provide the greatest separation in the data, which can be useful as an input into supervised learning algorithms.

The primary application of supervised learning algorithms is to make predictions on the labels of new data based on learned relationships in the training data. As with unsupervised algorithms, we can also consider an alternate interpretation of this information: given a prediction (and a prediction accuracy), what were the most important relationships in developing the prediction? Both making the predictions themselves as well as identifying the key features in obtaining accurate predictions could be useful in a human training setting. A brief summary of the general capabilities discussed here is presented in Table 3.3

Table 3.3: Capabilities of machine learning techniques

Machine Learning Technique	**Capabilities**
Unsupervised Learning	Identify natural groupings in the data Identify outliers in high dimensional space Select metrics for high data separation
Supervised Learning	Make predictions on labeled data Identify features important to accurate prediction

3.4.5 Potential Applications

In considering both the needs of human training and the capabilities of machine learning algorithms, a set of possible applications of machine learning to the human training domain can be proposed. These applications include both direct and indirect use of the capabilities of machine learning algorithms, and are described in detail below. A hierarchy of applications based on which types of algorithms are used in the application can be constructed and is shown in Figure 3-2. These are generally divided by whether they make use of unsupervised learning, supervised learning, or a combination of both methods. The following sections discuss each potential application in greater detail.

Improve Label Definiton

In a human training setting, defining the labels for supervised learning (that is, the performance scores of interest) is not always a trivial task. Consider the following scenario:

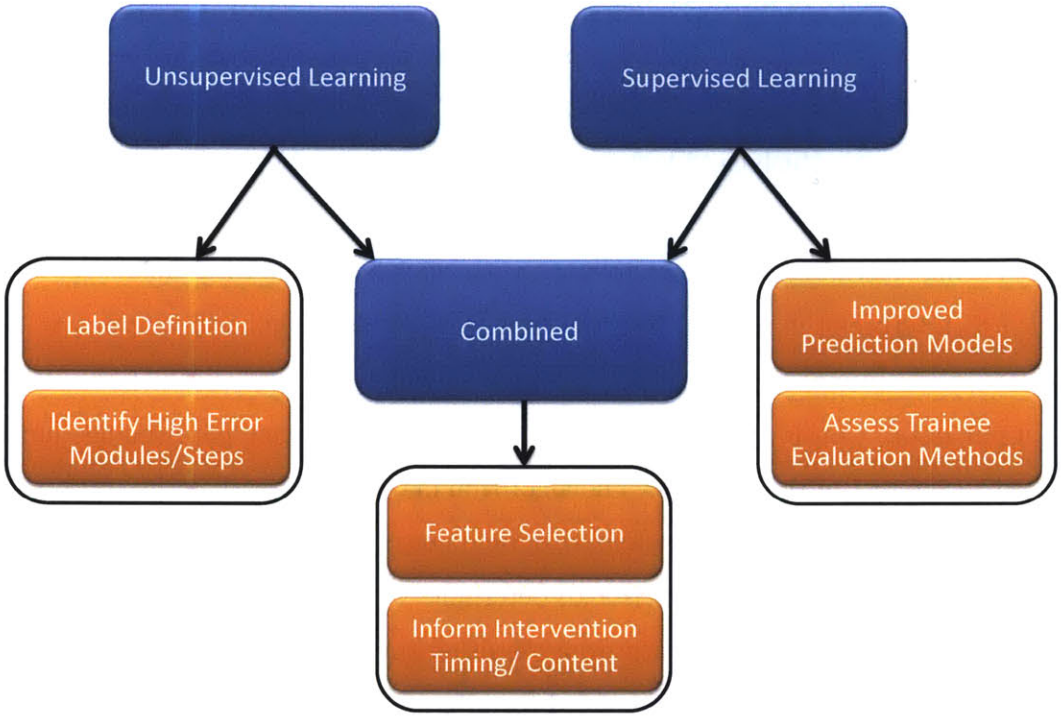


Figure 3-2: Hierarchy of Potential Machine Learning Applications to Human Training by Algorithm Type.

a training supervisor has received a set of performance scores from a group of trainees. As part of the assessment, the supervisor must separate trainees based on whether they have successfully completed the program or are in need of TI. Commonly, this ends up being a problem of needing to divide a continuous performance variable (or multiple variables) into a categorical classification, such as turning a numerical grade into a letter grade. Typically the trainer will use a set of cutoff performance values to determine which trainees fall into which category. Selecting these cutoffs can be challenging; one option might be to use subjective judgment of the trainees' performances. This could either be at a trainee-by-trainee decision level or subjectively deciding that a certain score determines the category of the trainee (e.g. >90% is an "A"). These types of determinations are simple to implement but as discussed in Chapter 2 have been challenged for their reliability [98]. Another option could be to use a subjectively-selected percentage-based determination, e.g. using three categories, and placing the top 1/3 in one category, the

middle 1/3 in another, and the bottom 1/3 in the last category. This strategy converts the data into an ordinal form, neglecting the actual performance values and instead relying on the performance of a trainee relative to his/her peers. A third option would be to objectively base the cutoffs on the data itself, finding natural divisions in the data that appropriately categorizes trainees. This falls directly into the capabilities of unsupervised clustering algorithms, which can identify natural groupings (and the splits between these groupings) using a data-driven approach.

When using clustering algorithms to assist with label definition, the human supervisor is needed to interpret the results. The machine learning algorithm will attempt to find the statistically optimal groupings of trainees and will do so regardless of the nature of the input data. Consider a case where there is a single extreme performance outlier compared to the rest of the trainees. The algorithm may determine that the statistically optimal grouping is to have the single outlier be its own group. Given this output from the algorithm, it is up to the supervisor to determine whether it is appropriate to treat that individual as a different category from the other trainees or whether to exclude the outlier from the analysis. Despite this need for interpretation of the results, for general human training datasets, clustering algorithms may be able to help objectively determine appropriate locations for the category cutoffs.

Identify High Error Modules/Steps

A module or step for which performance is poor across many trainees can have several interpretations. One possibility is that the module represents one of the more difficult topics in the training and results in greater performance separation between the strong trainees and the weak trainees. Additionally, by identifying the topics of greatest difficulty, it can assist with selecting and designing interventions for struggling trainees. Another interpretation is that the high error is created by the module not meeting its training objectives. This could be indicative that a redesign of the module is necessary to improve performance.

Based on these interpretations, the identification of topics or areas of high error for trainees can be useful to training programs in several ways. First, they allow for the identification of topics that may need intervention for the trainees. Second, they may be the most important predictors in identifying future performance. Third, high-error

modules may be indicative that the module is poorly designed. Each of these uses will be discussed further later in this chapter, but the remainder of this section focuses on the process of identifying these high-error modules. The identification of outlying values (such as "high-error" here) typically falls under the purview of unsupervised learning.

Under assumptions of the distribution, one can determine outliers based on heuristics of variance, such as $> 3\sigma$ for a normal distribution. It is easy to compute these measures for simple distributions, but as the complexity of the distribution and the dimensionality increase, statistical learning such as clustering can become a more efficient method. Using clustering algorithms can separate the continuous error data generated by modules or individual steps into categories based on the error. Clusters associated with large errors can be used to define the outlying "high-error" values. As with other applications of unsupervised learning, the algorithms are guaranteed to produce a cluster assignment for all data points, but the interpretation of the clusters as "high-error", etc. must be left to the training supervisor or training program manager.

Improved Prediction Models

The most direct application of supervised learning is for the prediction of labels on new data based on the feature values. In a human training setting, this involves using measures taken during the training process to make predictions of performance either later in the course or post-training. This has immense value to the human training domain, as was discussed in Chapter 2. Due to the investment of time and money into training programs, identifying those trainees that will not pass the program as early as possible can save resources for the organization.

Currently, many of the techniques used to estimate final performance are based on the simple assumption that future performance will be the same as past performance. An example from a knowledge-based environment would be using a mid-term grade as a direct prediction of final grade. Simple statistical learning models have also been previously utilized in this capacity. In the classroom, teachers have used techniques such as linear regression to make predictions on performance later in the course. These methods can identify basic trends in the data, and allow for the extrapolation about final student performance based on those trends. However, these techniques typically make strong assumptions about the data, such as a linear trend. More flexible supervised

learning techniques such as SVMs and ANNs could allow for fitting complex relationships in data but have not yet been applied in the human training domain.

There is a fundamental tradeoff between the accuracy and timing of these predictions in a human training setting. As more data is collected (which requires being later in the course), the accuracy of the predictions of final performance will increase. The most accurate predictions will include all data up until the end of the course. However, these predictions, while accurate, will have relatively little utility—having a prediction of the final performance at the end of the program does not allow for intervention to improve struggling trainees or attrition that will save the program resources. Therefore, an appropriate balance must be found between accuracy of the prediction by the additional collection/use of data and the emphasis on trying to make predictions early on in the program. This tradeoff will be further discussed in the analyses of the example datasets in Chapters 4 and 5.

Assess Trainee Evaluation Methods

An additional retrospective analysis can be performed using supervised learning algorithms that is of use to training programs. As discussed previously, there is effort and cost committed in collecting data metrics for use in training evaluation, and the collection of data is sometimes disruptive for the trainee [82–88]. Therefore, it seems logical that a training program would want to ensure that the metrics used have a high efficiency, that is, high utility of information gained relative to the effort spent in collection. On the reverse side, a program would like to limit the number of inefficient metrics. To determine a metrics efficiency one has to be able to both quantify the benefit of the metric as well as the cost of collection. Supervised learning techniques can assist with the former task.

When a supervised learning algorithm is applied to a set of data, not only are the parameters of the model learned, but the user gains some insight into how important each of the inputs are in the final model. The specifics will vary by the machine learning technique used, but typically there is a learned weighting of the inputs involved in the creation of a prediction of the output. If a weighting is small, this implies that the input is not heavily used in the prediction, whereas if it is large this input plays a major role in the prediction. In this way, the training program can obtain an objective assessment of the impact of each input metric. This represents a quantification of the benefit of

the metric as determined by its usefulness in a predictive capacity and thus useful for determining whether each metric justifies the cost of its collection. Consider two features that are used as part of a larger model in a regression analysis. If the removal of one feature results in the reduction of the model R^2 by 0.01, this feature is not providing much information to the model. If the removal of the other feature results in a reduction of 0.4, this feature is important for model accuracy and should likely be kept in the training program for its predictive value.

Feature Selection

As discussed above, unsupervised learning methods can assist in the identification of features that provide the greatest separation in the data. An additional use for this strategy is to utilize the highest variance features as an input into supervised learning algorithms (i.e. dimensionality reduction). Supervised learning algorithms (on both classification and regression problems) require some amount of variability in the data along each feature to be able to discriminate between data points. If a relationship exists (such that labels are dependent upon features), having a high level of discriminability on the features will allow for a more accurate model.

In a human training context, this means that we can downselect from the initial large set of features to utilize the features that are most useful for the supervised learning algorithms. While it is difficult to improve accuracy over the "kitchen-sink" approach that uses all possible inputs, developing a parsimonious model that focuses on the most important aspects can maintain a high level of accuracy while requiring less effort to train the algorithm and improving the ease of the interpretation of results. This type of feature downselection is particularly important in the human training domain. As previously discussed, the number of data points (trainees) may be relatively limited and in machine learning it is important to have a greater number of data points n than features m ($n \gg m$, dependent on the signal-to-noise ratio of the data).

The reason for this becomes apparent in considering general modeling in high dimensional space. If there are as many dimensions as datapoints, the model can simply fit the weight for each dimension to capture one data point, and ignore the other features. This produces a trivial result that is a perfect fit in training the model, but tells the user little about how each feature is useful in the prediction. Additionally, this represents a

major overfitting of the data, and will result in poor generalization performance. Thus, the ability to identify and downselect appropriate features for use in supervised learning will provide models that are both relatively accurate and generalizable.

Inform Intervention Timing/Content

An important human training application of these methods involves combining two of the previous applications: identifying high-error modules and improved prediction models. If we have an accurate early prediction that a trainee is likely to fail the program, it is important to consider possible interventions that would either remove the trainee from the program early or to provide retraining of some of the critical material. As discussed in the high-error module section, one interpretation of a high-error module is that it highlights the most difficult material in the course. These areas can provide topics for the training supervisor to review with a struggling trainee. By combining both unsupervised and supervised approaches, we can identify the "who," "what," and "when" for intervention: the trainees that are predicted to fail, the difficult material, and the earliest that a high accuracy prediction can be made.

Additionally, we can take an individualized intervention approach by using supervised learning as a flag. If a trainees prediction of failure reaches a certain accuracy, this would indicate need for TI. By looking back at the specific errors on modules committed by that trainee, the training supervisor can get some idea of the areas that were of great difficulty to that individual trainee.

3.5 Chapter Summary

This chapter addressed the considerations in the applications of machine learning approaches to assist with training evaluation. First it discussed the properties of training data in greater detail, particularly focusing on data from rule-based and knowledge-based environments. Next, several example environments were presented to frame the discussion of machine learning approaches. Third, the implications of training data on machine learning algorithms were discussed, including options for data standardization, distance measures, and metrics for model evaluation and comparison. Last, the capabilities of machine learning algorithms were discussed in the context of training program

needs, resulting in a set of proposed applications of machine learning to training datasets. Empirical results for these applications for both knowledge-based and rule-based environments are presented in Chapters 4 and 5, respectively. After these results are discussed, Chapter 6 presents a comparison of the machine learning results across these two training domains.

Chapter 4

Application of Machine Learning to Rule-Based Training Environments

Chapter 2 presented a taxonomy of training domains, generally dividing training based on the task type required by the operational environment: skill-based, rule-based, and knowledge-based. These task types are listed in increasing order in terms of the cognitive complexity associated with the tasks; skill-based tasks typically require little to no cognitive effort, rule-based tasks require some effort, while knowledge based tasks require deep cognitive understanding to complete. With higher cognitive complexity, greater variation and novelty of the assessment metrics are expected, which may create challenges for machine learning applications to these datasets.

Therefore, the first training environment selected for analysis (presented in this chapter) is representative of a supervisory control rule-based environment. Machine learning analysis on the higher cognitively complex knowledge-based training dataset is presented in the following chapter, Chapter 5.

In rule-based environments, there is a heavy focus on repetition of tasks during training to achieve a high level of familiarity and expertise in executing the procedures used in the task. Unlike training for knowledge-based settings where assessments may involve situations that have not been previously seen by the trainees, rule-based training will often use the same tasks in training as seen in the operational environment. This seems particularly logical in cases where the trainees will be performing the task regularly in an operational setting, such as the shutting down of an engine. However, this strategy also impacts the nature of the data collected during training assessment, which must be con-

sidered during the interpretation of trainee performance and errors. As computer-based training (CBT) technologies are incorporated into rule-based settings, opportunities for higher frequency assessment and new metrics will create larger and more complex datasets that suggest the use of machine learning approaches. In this chapter, a dataset representative of a rule-based training setting is used to test the application of machine learning for informing training evaluation.

This chapter is divided into five sections. The first section presents a brief discussion of the rule-based environment chosen for this analysis: nuclear power plant operation. The second section provides a brief overview of the typical assessment techniques used in rule-based environments, and the expected set of features that would be available for machine learning analysis. The third section presents the details of a data collection experiment that incorporates a range of trainee assessment metrics that would typically be used in rule-based CBT environments. This section also specifies the features and targets to be considered for machine learning approaches. The fourth section discusses the methods and results of both unsupervised and supervised learning on this rule-based dataset. The final section of this chapter provides some lessons and conclusions from the analysis.

4.1 Nuclear Power Plant Operation and Training

Training for nuclear power plant operation was selected as a representative supervisory control rule-based environment for the collection of a training dataset. Rule-based environments such as the operation of nuclear power plants are characterized by their use of procedures and heuristics in the completion of tasks. In nuclear power plants, the procedures act as a guide for operators to monitor and control the plant. These procedures are typically paper-based, and necessarily complex to address problems that may arise with the complicated plant system [146]. In modern nuclear power plants, operators utilize these procedures to complete both normal (Standard Operating Procedure, or SOP) and off-nominal operations. The operators are expected to follow the procedures as they monitor and operate the plant. Thus, this environment typically makes the assumption that the procedures are well designed and can be used to address any issue. If the procedures are considered to be the "correct" path, then adherence to the procedures will relate to

improved performance and safety on the job. Therefore, training operators to use and maintain procedure adherence skills is of great interest to training programs in these environments. It should be noted that there may be cases of poorly designed procedures that do not represent an optimal action path. However, investigating the optimal design of procedures is beyond the scope of this thesis, and thus in this analysis the adherence to procedures is assumed to indicate strong trainee performance.

As discussed in Chapter 3, training for these rule-based environments typically focuses on a high level of repetition to prepare for the operational environment. For rule-based environments such as the nuclear reactor dataset presented in this chapter, assessment metrics typically include subjective and objective metric types. Subjective metrics are currently widely used in domains such as nuclear power plant operation, and include ratings given by both the training supervisor as well as self-ratings by the trainees. Objective metrics include performance data such as the power output of the reactor during a training module or other assessments such as written examinations.

The use of computer-based training (CBT) technologies allows for operator practice in a simulated environment of the real-world system and the potential development of a wide range of training scenarios. Because the following of procedures is particularly important in rule-based settings, the ability to monitor the process-level data of the trainees as they perform each action on the system allows for the collection of new types of assessment data that measure the procedure adherence of the trainees. While not currently used in rule-based training assessment, these metrics are enabled by the use of CBT technologies and provide new opportunities for training evaluation. Procedure adherence metrics are also objective by nature but are separately discussed here since they are metrics that are specifically enabled by the use of CBT environments. Each of the three types of assessment information from rule-based environments (subjective, objective, and procedure adherence) are gathered in the collected dataset and compared in the machine learning analysis later in this chapter.

Prior to the presentation of the specific data collection methods for the rule-based dataset, it is useful to consider the properties of data collected from each of these three types of assessment metrics. Understanding the types of data that arise from rule-based environments is important not only to frame the data collection experiment, but also informs the selection of machine learning algorithms and the interpretation of results.

4.2 Features in Rule-Based Training

Currently, environments such as the nuclear reactor operator training often utilize subjective assessment as an important part of determining trainee progress. Typically, there are two main types of subjective assessment: self-assessment and supervisor assessment. Self-assessment requires the trainee to provide their own perceptions of performance and progress, and could be collected at the end of each training module or at the end of the program. Supervisor assessment provides similar ratings of performance, based on the experience and knowledge of the instructor. Both self- and supervisor assessment are often gathered either as ratings (such as on a Likert scale), or through expository writing. The latter requires significant interpretation, and would be difficult to include as features in a machine learning approach. Subjective ratings, on the other hand, can be readily incorporated as discrete numeric features in machine learning by using the integer values (e.g. 1-5 for a 5-point Likert scale).

Objective performance metrics may also be collected during rule-based training, and will highly depend upon the training domain. Traditional non-CBT objective metrics usually represent summative performance at the level of a module, such as whether or not the trainee successfully resolved an emergency. CBT training in nuclear reactor operation typically relies on a simulation of the reactor system which allows the trainee to practice in a realistic setting under a variety of conditions. Use of a simulation allows trainees to encounter situations and perform actions that dramatically affect the performance of the reactor. Thus, measurements of the reactor performance (such as power output or length of time reactor is inactive) represent objective metrics that can be collected during the training program. While these are not always collected in current training programs, the use of CBT allows easy collection of these measures. As mentioned in Chapter 3, since these metrics can vary widely in numeric range, the use of data standardization techniques may be particularly important for objective performance metrics.

Procedure adherence metrics are also enabled by the use of CBT in rule-based environments through the gathering of individual actions or sets of actions. Two types of procedure adherence are of interest in nuclear reactor operation: the ability to identify the correct procedure (termed "procedure selection"), and the fidelity to which the procedure is followed (termed "adherence"). The correct selection of procedure could simply be expressed through a binary of whether the trainee did ("1") or did not ("0") choose the

correct procedure. However, there are no current standard methods for the measurement of adherence, and thus it is worth considering a range of potential adherence metrics. Appendix Q provides a detailed discussion of possible measurements of adherence, and the selection of appropriate adherence metrics that could be used as features in machine learning analysis.

The analysis in Appendix Q establishes that a procedure can treated as a sequence, and comparing the set of steps the trainee actually performs to the set of intended steps can be accomplished through sequence distance techniques. Three potential sequence distance metrics were compared as possible adherence metrics: two sequence-based metrics (Levenshtein distance and suffix arrays), and a novel model-based metric (Procedure Adherence Metric, or PAM). The Levenshtein distance directly measures trainee errors through the number of additions, omissions, and substitutions. Suffix arrays measure similarity between trainee actions and intended actions through the longest matching subsequence. The PAM uses action frequencies to provide a measure of "correctness" of the trainee sequence. A theoretical comparison indicated that the PAM may be more sensitive to the order of actions, while the Levenshtein distance has greater ease of interpretability (see Appendix Q for details). Therefore, both the PAM and Levenshtein distance metrics are considered as possible adherence metrics in the machine learning analysis presented later in this chapter.

Adherence metrics are dependent upon the individual actions that the trainee performs, and thus represent the detailed "process-level" information discussed in Chapter 1. Specifically, by measuring adherence at each action, these metrics monitor not just whether an overall task was completed but whether the individual actions taken to complete the task was correct. These metrics can also be combined into a module-level "summative" form, that tracks the overall adherence for each module. In the analysis presented later in this chapter, the value of the process-level and summative forms of adherence measurement are compared. The process-level metrics at any particular action can be calculated for Levenshtein distance and PAM using Equations C.1 and C.6 in Appendix C, respectively. The summative form uses the same calculation for the entire trainee and prescribed sequences for the module.

A representative dataset for testing machine learning approaches should incorporate all three of these types of data: subjective, objective, and procedure adherence metrics.

The next section describes the collection of an example dataset that contains all of these elements, for use as features and targets in machine learning presented later in this chapter.

4.3 Data Collection Experiment

An experiment to collect an example rule-based training dataset was conducted using a simulated nuclear power plant computer-based training (CBT) interface. The training program consisted of a series of four modules in two phases: a training phase with three training modules and a test phase with a single test module. The training modules were constructed with generally greater guidance than the test module, and were intended to familiarize the trainees with various aspects of power plant operation and emergency resolution. The test module was designed to mimic an operational setting, where the trainees acted as operators monitoring normal reactor operation and were responsible for solving any problems that arose. In both training and test modules, participant were required to utilize different sets of procedures to complete the modules; that is, both procedure selection and adherence were important to completing the modules. The full procedures used in the experiment are included in Appendix E.

Training modules were labeled T-1, T-2, and T-3, and designed to be of increasing difficulty as the participants progressed through the training program. T-1 introduced the participants to the reactor interface (shown in Figure 4-1) and walked them through the process of a shut-down and start-up of the reactor. T-2 introduced the participants to the controls for the coolant system on the reactor and had a series of goals that the participant needed to achieve though the use of coolant system controls. It further challenged participants by presenting them with an initial reactor state that did not match the conditions required for starting the training module. Participants had to realize the discrepancy without it being explicitly pointed out and correct it before starting the training module. T-3 provided instruction on how to diagnose a problem with the reactor and the use of emergency procedures. The actual problem was unknown to the participants at the beginning of the module, and over the course of the module the participants diagnosed the issue from one of several possible diagnoses through the use of the procedures.

The procedures were designed in a two-column format based upon Nuclear Regulatory Commission (NRC) procedures. Figure 4-2 shows an example page from the procedures. These columns are entitled "Action / Expected Response" and "Response Not Obtained". The Action / Expected Response column specifies the action for the operator to perform to complete that step. If there is a condition that must be satisfied (such as a check that a pressure is below a particular value), this value is also specified in the Action / Expected Response column. After completing the action for that step, the operator either continues to the next step if the expected conditions are met or performs the actions in the Response Not Obtained column if the conditions were not. As can be seen in Step 8 of the procedures in Figure 4-2, the Response Not Obtained column might also specify moving to a different procedure (in this case E-2).

A single test scenario placed participants in a seemingly normal operational scenario in which they were instructed to note any problems and respond accordingly. Participants were tested in their ability to perform tasks similar to what they had done in

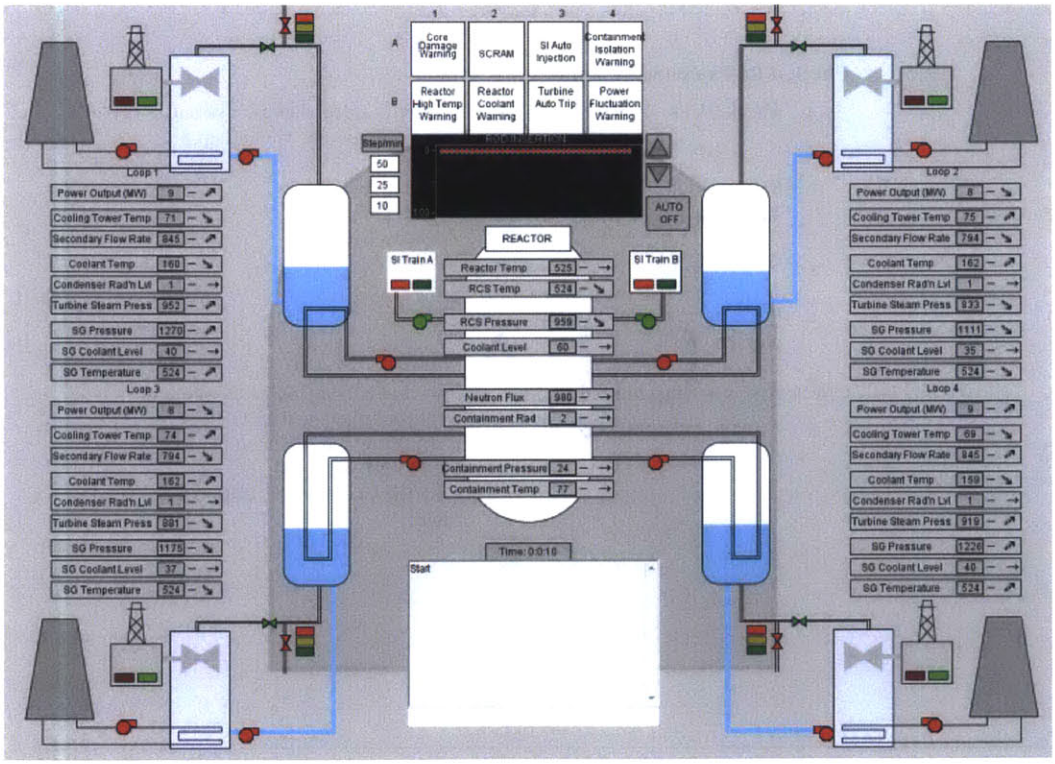


Figure 4-1: Simplified nuclear reactor interface used in data collection experiment

E-0 Reactor Scram or Safety Injection

STEP	ACTION / EXPECTED RESPONSE	RESPONSE NOT OBTAINED
5	**Verify Secondary Coolant Flow for ALL Loops – GREATER THAN 720 GPM**	IF flow less than 720 GPM, THEN: a. Ensure secondary coolant pumps are activated b. Stop dumping steam
6	**Check RCS Temperature:** • IF any RCP running, THEN check RCS average temperature – STABLE BETWEEN 557° AND 562° OR • IF no RCP running, THEN check reactor coolant temperature – STABLE BETWEEN 557° AND 562°	IF temperature less than 557° and dropping, THEN: a. Stop dumping steam b. IF cooldown continues, THEN close main steamline valves IF temperature greater than 562° and rising, THEN open ALL steam dump valves
7	**Check if RCPs should be stopped:** a. Check RCPs – ANY RUNNING b. RCS pressure – LESS THAN 1360 psig c. Stop all RCPs d. Place steam dumps in steam pressure mode	a. Verify steam dumps in steam pressure mode. Go to Step 8 b. Go to Step 8
8	**Check if SGs are not faulted:** a. All SGs - PRESSURIZED	IF pressure in any SG is dropping in an uncontrolled manner OR any SG is depressurized, THEN go to **E-2, FAULTED STEAM GENERATOR ISOLATION, Step 1**

5

Figure 4-2: Example procedures from nuclear reactor training data collection experiment.

the training modules in a less structured format that required procedure selection in addition to procedure adherence. For the operational test module, participants were provided a binder similar to those typically used in nuclear reactor operations, containing maintenance procedures (M-1), emergency procedures (E-0, E-1, E-2, E-3), anomalous operating procedures (AOP-1, AOP-2, AOP-3, AOP-4, AOP-5, AOP-6), and several appendices containing nominal ranges of values for the reactor components (see Appendix E for full procedures).

4.3.1 Interface

The interface used in the experiment was a simplified nuclear power plant interface which contained four power-generating loops (see Figure 4-1). Users could control different subsystems of the plant, including pumps, valves, turbines, safety systems, and control rods. Displays on the interface indicate different states of the four power-generating loops, the reactor, position of the control rods, and various alarms which may be activated during abnormal states of the reactor. The lower-central part of the interface contains a text chatbox with which the user may be given updates or instructions. These instructions took the form of pre-programmed responses in the interface software. Users could also refer to a reference screen which provided labels to all the system elements. The same interface was used for both training and test modules.

Participants interacted with the interface through the computer mouse and keyboard. To change system states (such as turning on a pump), the participant could left-click on the desired element in the interface. Some procedures also called for "verifications" which indicate that the operator has noted a particular system value, such as verifying a pressure reading. To represent the deliberate checking of these elements, participants were instructed to right-click on elements that were being verified. Additionally, certain procedures required the use of the chatbox to report particular system values or request maintenance services, and the keyboard was used to enter these messages.

4.3.2 Participants

The experiment participant pool consisted of 47 individuals, 21 females and 26 males, ages 18 to 43. The average age was 21.64 with a standard deviation of 5.53. participants were recruited by posters, emails, and by word of mouth. Participants were paid $100

for completion of the experiment. See Appendix L for the consent form used in the experiment.

An initial demographic survey revealed that four participants had experience with nuclear power systems, none had colorblindness, and all had 20/20 or corrected-to 20/20 vision. Two percent of participants played video games daily, 10% played a few times a week, 14% played weekly, 32% played monthly, and the rest rarely played. See Appendix M for the survey used in the experiment.

4.3.3 Task

Training Phase

During the training phase of the experiment, participants were instructed in the use of the nuclear reactor interface and allowed to interact with the simulation as part of that training. First, a PowerPoint introduction to the nuclear interface and the operation of the reactor was given (see Appendix N for the tutorial slides used). Participants were allowed to go through the slides individually at their own pace. After the introduction, the participants began the three training modules. Each training module was designed to last no more than 30 minutes. Each module consisted of:

- Watching a narrated video showing the reactor interface and explaining the procedures involved in that particular module. The videos became progressively less specific; the first guided the participant through all the steps, while the latter two increasingly focused on the general principles behind the procedures.
- Carrying out the training procedures using the interface by following a set of instructions. Instructions used in these modules were based on the Nuclear Regulatory Commission's standard operating procedures for nuclear power plants.
- Rating the module on procedural clarity and difficulty of the module.
- Taking a brief multiple-choice quiz on the procedure they just performed to assess their understanding of the module.

This video, simulation, rating, quiz sequence was repeated for each of the three modules (see Appendix O for the quizzes used in the experiment). During this training

period, participants were allowed to ask the experiment supervisor for clarification of the interface or procedures, but were not provided help on the quiz. All participants went through the same training procedure and were given the same quizzes. Quiz solutions were provided after the completion of each quiz to ensure that all participants had the same information before going into the following module.

Test Phase

During the test phase of the experiment, participants were given a set of operating procedures consisting of emergency operating procedures, anomalous operating procedures, maintenance procedures, and reference appendices. They were instructed to act in the role of a nuclear reactor operator doing routine supervision of the reactor, and monitor the operation of the plant for any unusual activity and carry out instructions given to them via the communications chatbox. Five minutes into the test phase, a steam valve leak was simulated as a system anomaly, resulting in a gradual drop in performance of one of the loops. Successful resolution of the scenario involved diagnosing the problem and using the correct Anomalous Operating Procedure to attempt to solve the problem, followed by use of a separate Maintenance Procedure to complete the solution. All participants were given the same problem in the testing scenario. Participants were given a 90-minute time limit to complete this scenario. No assistance was provided to the participants by the experiment supervisor during the test phase.

Data Collection

Each click on the interface was recorded in a log file along with an identification code and time stamp. Since participants' observation of reactor status displays was not directly recorded, participants were asked to note the completion of "verification" actions by right-clicking on a display panel or control, rather than the normal left-click used to interact with controls. Ineffective actions such as clicks on inactive parts of the interface were also recorded, along with interactions with the chatbox and use of the help screen. A screen recording of each session was also taken as a separate record of the participant behavior. A survey of the participants was also taken after the experiment that was primarily a self-assessment of performance in the experiment, but also allowed for comments about the interface and procedures to be given. The post-experiment survey is included in

Appendix P.

The assessment metrics that were considered for use as features and targets in machine learning approaches are listed in Table 4.1. As discussed earlier, these features are generally divided into categories of subjective, objective, and procedure adherence. Many of the features apply to all modules, but several additional features were designed into particular modules and can also be used in machine learning approaches. The exact number of features for process-level adherence metrics depended upon the number of actions in each module. The summative adherence features shown in Table 4.1 refer to the adherence value at the end of each module, rather than action-by-action. In the machine learning analysis presented later in this chapter, the assessment metrics collected during the training modules were used as features while the metrics during the test module were used as targets. It may be noted that the post-experiment survey is only available after the completion of the training program, and is therefore only appropriate to include in post-hoc predictions during supervised analysis and unsupervised learning such as clustering.

While there are an extremely large number of variations on many of these features (such as the measurement choice for procedure adherence), this list contains a selection representative of the different types of data gathered during the data collection experiment. With the dataset collected and the potential features identified, the next section presents the results of both unsupervised and supervised machine learning algorithms on this dataset.

4.4 Methods and Results

As discussed in Chapter 3, both unsupervised and supervised machine learning algorithms have potential benefits for training evaluation in procedure-based training environments. As a reminder, unsupervised algorithms are a descriptive approach that identifies relationships and natural groupings that exist within a dataset. This could be useful for identifying trainees that exhibit similar behaviors to each other or topics/modules that are of particular difficulty for the trainees. Additionally, feature selection methods can reduce the dimensionality of the feature space while attempting to maintain the most relevant information in preparation for supervised learning methods. Supervised learning

Table 4.1: Features and Targets from Nuclear Reactor CBT experiment

	Name	Type	Dimensions	Description
General Features	Levenshtein Distance	Adherence	372	Process-level Levenshtein distance at each action during training modules
	Levenshtein Distance (summative)	Adherence	3	Final Levenshtein distance for each training module
	PAM	Adherence	372	Process-level PAM at each action during training modules
	PAM (summative)	Adherence	3	Final PAM for each training module
	Quiz scores	Objective	3	Score for each post-module quiz based on number of correct questions (0-5 or 0-6)
	Demographics	Objective	10	Various demographic information, such as age, gender, and experience with procedures
	Module Ratings	Subjective	6	Ratings of procedural clarity and difficulty for each training module
	Post-Experiment Survey	Subjective	7	Ratings of overall confidence, reactions, and procedural clarity
Module-Specific	Initial Condition Check	Objective	1	Binary of whether trainee identified initial condition deviation in Module 2
	Correct emergency diagnostic	Objective	1	Binary of whether trainee correctly diagnosed emergency in Module 3
Targets	Levenshtein Distance (summative)	Adherence	1	Final Levenshtein distance for test module
	PAM (summative)	Adherence	1	Final PAM for test module
	Correct Procedure Selection	Objective	1	Binary of whether trainee selected correct procedure first
	Completed Solution	Objective	1	Binary of whether trainee completed all parts of the problem solution
	Reactor Power Output	Objective	1	Average power output from reactor over test module

can be useful as a predictive approach to training evaluation, when labels are defined to be trainee performance metrics at a later point in the training program. This can be useful in a post-hoc approach to identify assessment metrics that contribute to prediction performance, and also can be used partway through a training program to determine which trainees may need early intervention.

4.4.1 Unsupervised Learning

As discussed in Chapter 2, unsupervised learning techniques can generally be divided into clustering and feature selection methods. Clustering techniques find natural groupings in the data based on similarities across the feature set. In training evaluation, clustering algorithms can help to identify groups of trainees with similar behavior and then used to select a subset of trainees that may all benefit from the intervention. Additionally, clustering algorithms can assist with the selection of labels when converting a continuous performance metric (such as power output) into a discrete category (such as "good" and "poor") to further classify performance.

Feature selection methods analyze the features themselves to identify useful inputs to machine learning models or create new inputs (through combination or transformation) that describe the variability in the data with as few features as possible. This is particularly important for the rule-based dataset described here, due to the large number of features available through the process-level information in comparison the number of trainees. To prevent overfitting during supervised learning, dimensionality reduction (a form of feature selection) can be utilized to condense the features into a smaller feature space. In this section both clustering and feature selection methods are applied to the rule-based dataset, and a range of algorithms are tested for their usefulness.

On the rule-based training dataset presented in this chapter, there is no *a priori* information on the usefulness of particular features. Therefore, it is appropriate to take an iterative approach to unsupervised learning and feature selection. Sets of features can be tested in the algorithms, and then refined to identify the features and feature types that provide the best machine learning model performance. In later sections, this approach is taken for selecting appropriate feature sets to be used in both unsupervised and supervised learning, by comparing performances between adherence metrics, across adherence, objective, and subjective metrics, and investigating the usefulness of dimen-

sionality reduction on process-level features.

Clustering Algorithms

A selection of common clustering methods of varying complexity was presented in Chapter 3, and the same set was applied to the rule-based dataset. As a reminder, the methods include k-means, hierarchical clustering, Gaussian Mixture Models (GMMs) and Self-Organizing Maps (SOMs). These methods represent a range of clustering algorithm types: a centroid model, connectivity model, distribution model, and topological model, respectively. An analysis of these methods on the nuclear reactor CBT dataset described here provides insight into which methods (if any) are useful on data from a typical rule-based training domain.

As discussed in Chapter 3, there are several necessary preparatory steps prior to the use of clustering approaches. These include any necessary data pre-processing (including standardization), selecting an appropriate distance measure, and identifying appropriate metrics for cluster algorithm comparison. These topics are discussed in the following sections.

Data Standardization and Pre-Processing

The features included in the rule-based dataset have a wide range of measurement scales, from binary measures (0 or 1) to continuous measures that range from zero to values in the hundreds. Thus, data transformation is critical to ensure that each feature is able to be treated approximately equally during clustering. To empirically determine the best transformation, both z-score and range transformations were tested for clustering performance using a basic k-means algorithm across all features from Table 4.1 (see Appendix F for full results). For most feature sets, the cluster algorithm performance using a range transformation outperformed those based on a z-score transformation. This is likely attributed to the lack of normally-distributed data along most features, particularly the procedure adherence features. In particular several trainees had a tendency to become lost in the procedures, and resorted to seemingly random interactions with the interface. These cases resulted in particularly large adherence deviations compared to trainees who tended to stay adherent to the procedures even after slight deviations (creating a bimodal distribution). Based on these results, the range transformation was selected for use in

the subsequent clustering analyses.

For this rule-based dataset, a large number of features are available, as can be seen from Table 4.1. This presents a major concern for unsupervised learning techniques as discussed in Chapter 3 (the "curse of dimensionality"). As a reminder, this can present challenges to cluster algorithms by reducing the discriminability of distances between points. To address this issue, only summative (end of module) adherence metrics were used in the cluster analysis. While this has the disadvantage of being unable to group trainees based on single actions taken during the modules, it dramatically reduces the dimensionality of the feature space and allows for a more meaningful measurement of cluster distances.

Distance Measure

The set of potential distance measures presented in Chapter 3 were considered for this dataset: the Euclidean distance, Manhattan (or city-block) distance, and the Chebychev distance. The nature of the features included in the analysis has a strong impact on the selection of an appropriate distance measure, and the nuclear reactor CBT dataset has a wide range of feature types. As seen in Table 4.1, these generally fall under the categories of adherence, objective performance, and subjective performance. Since a distance measure should be consistent for the entire feature space, it is important to consider all these feature types in the selection of an appropriate distance measure.

Adherence data are measured on a scale of 0 to ∞, where 0 represents no deviations from the prescribed procedures. The maximal deviation is potentially infinite, as the trainee could continue to add actions far beyond the intended termination of the procedure. Consider a module using a procedure with 30 intended actions. If a trainee actually performs 100 actions while completing this module, the minimum Levenshtein distance for this trainee is 70, even if all of the additional actions are repeats of correct actions on the procedure. These features provide the potential for large differences between trainees which has important implications for the Chebychev distance, which identifies the largest difference along any feature between trainees. While adherence may be a reasonable way to differentiate or cluster trainees, Chebychev distance might do so at the exclusion of other metrics such as objective assessments. For Euclidean and Manhattan distance, the main differences will depend upon the number of features. Specifically, Manhattan dis-

tance increases linearly with the number of features, while Euclidean distance depresses the impact of each feature on distance as more features are added. By considering only the summative adherence metrics in clustering, either the Euclidean or Manhattan distance is appropriate for these features.

There are several objective performance metrics collected in this dataset, including the quiz scores, module-specific items such as correctly diagnosing problems with the system, and the power output from the reactor during the test module. These features vary in their ranges of values: quizzes were scored based on the number of questions correct (0-5 or 0-6), power output ranged from 0-34, and other features such as correct diagnostics were recorded as a binary (either 0 or 1). Unlike adherence metrics, these objective metrics are all bounded with maximum and minimum values. This creates an interesting contrast with the adherence metrics, which are theoretically unbounded. Since Chebychev distance will select the highest difference among all features, it is inappropriate to use this distance on features generated under radically different processes (such as bounded vs. unbounded metrics). With only a few features, there will likely be little difference between Euclidean and Manhattan distances for these metrics. The subjective performance features, which included the module ratings and post-experiment survey (measured on a 5-point Likert scale) will exhibit similar properties to some of the objective metrics and have the same concerns with using Chebychev distance. They will also similarly have little expected difference between Euclidean and Manhattan approaches. Given that the differences between these distance measures are likely small for the types of data in this dataset, the more common metric of Euclidean distance was selected for clustering approaches.

Weighting of features could also be used on this dataset if it was determined that certain features were more important than others in determining similarity of trainees. However, there is not an obvious weighting strategy given the wide range of feature types in this dataset. It is likely that the summative adherence metrics are of higher importance to training supervisors than the individual process-level metrics, but choosing a specific weighting strategy would require a subjective assessment from the supervisor, and it is not clear in these early stages of this research whether such assessments are accurate. Additionally, in any procedure there are typically steps or actions that may be more important than others relative to system performance or safety. For example, the action of lowering the control rods when the reactor is in an emergency is likely

more important than checking the temperature of a steam generator. In fact, it is known that expert operators often skip actions that they have learned have little impact on the task [19, 80, 81]. However, the relative "importance" of any particular step or action is typically a subjective judgment of the subject matter expert (SME), such as an operator or supervisor. This then provides little guidance into the specific weighting to be used in the calculation of distances between trainees, and the quantification of the importance of individual actions is beyond the scope of this thesis. Thus, for the purposes of this analysis, the standard Euclidean distance was used rather than a weighted form. However, it should be noted that future work could make use of SMEs to attempt to quantify weights for individual actions in procedural training settings.

Metrics for Cluster Algorithm Comparison

To compare the algorithms, a set of measures of performance must be selected. As discussed in Chapter 3, both external and internal metrics can be used. As a reminder, external metrics utilize labels as in supervised data to indicate "true" clusters, while internal metrics provide performance characteristics about the clusters themselves. In this analysis, the external metrics of entropy and purity as well as the internal metrics of compactness and isolation are used. Both summative adherence and objective targets from the test module as shown in Table 4.1 were used for the calculation of external metrics.

Prior to any cluster analysis, a set of features must be selected as the inputs to the algorithm and the number of clusters must be chosen. For the rule-based training dataset presented here, there is not a clear selection for either of these requirements. Specifically, ideally features for clustering should provide useful separations of the data, and without initial testing it is unclear as to which training metrics collected in this dataset provide strong clustering performance. The selection of the number of clusters is likewise difficult on this dataset. This selection could be done subjectively, such as a training evaluator that would like trainees to be split into a certain number of groups. Sometimes the domain suggests a certain number of performance categories, such as in education when using a letter grading scale ("A","B","C","D","F"). On this rule-based dataset, such a natural division is not clear.

This issue reveals some of the fundamental iterative nature of clustering on an un-

familiar dataset. To determine the ideal features for clustering, a particular number of clusters must be selected. Likewise, to identify the appropriate number of clusters, a certain set of features must be used in the analysis. Therefore. the general strategy of analysis presented below represents a series of analyses to identify both an appropriate number of clusters and identifying ideal clustering features. The following sections describe a sequence of four analyses:

1. Identify number of clusters based on final performance
2. Given a number of clusters, determine which adherence metric (Levenshtein or PAM) provide better clustering performance - only one adherence metric should be used in clustering and supervised learning as they contain redundant information
3. Given an adherence metric, compare cluster performance between adherence, objective, and subjective metrics
4. Compare clustering performance of process-level adherence features to summative adherence to determine the advantages of temporally-based data collection on cluster algorithm performance

While a different starting point could have been selected for these analyses, this represents a logical flow for the identification of both number of clusters and the identification of features that provide high separation of data in the feature space. These investigations could be performed in an iterative fashion to isolate the best features and parameters for both unsupervised and supervised learning, as shown in Figure 4-3. After selecting a number of clusters, a series of feature selection analyses can identify which features provide the best clustering performance. Note that since strong cluster performance is indicative of good separation in the data, these analyses also suggest the use of these features for supervised approaches. Once a set of features have been selected, the analyst can return to the original assumption on number of clusters and repeat the process. In this chapter, one iteration of the process is presented for brevity, beginning with the selection of the number of clusters.

Selecting number of clusters

For the purposes of identifying an appropriate number of clusters on this dataset, using final performance to select an appropriate number of groupings is an intuitive place to

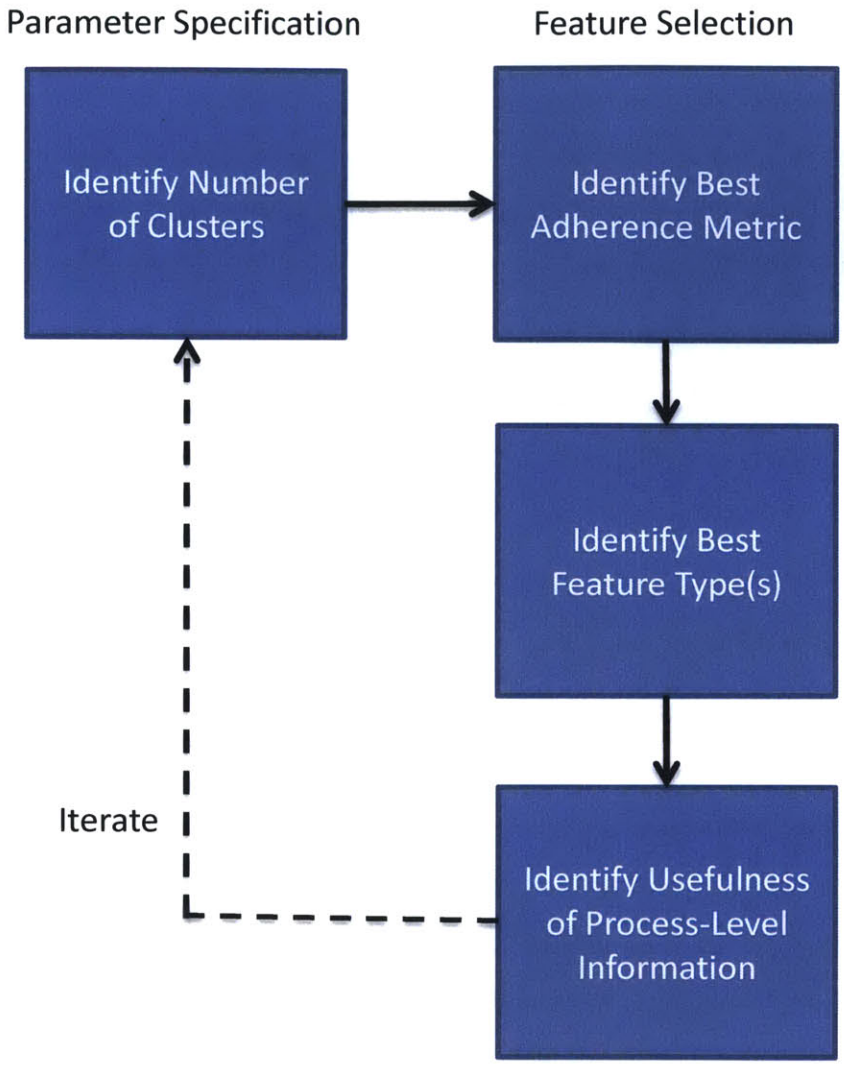


Figure 4-3: Schematic of Iterative Approach to Unsupervised Learning On Rule-Based Dataset.

start. This has the advantage of carrying more meaning for training evaluators; clusters can be associated with performance categories (e.g. "good", "poor"). While this could be subjectively selected based on the needs of the evaluator, it can be objectively determined through the use of model selection techniques. In model-selection, a set of cluster models are created with differing numbers of clusters, and the best model is identified through a chosen criterion.

The most common criterion for selecting the number of clusters uses the Bayesian Information Criterion (BIC) [12]. The BIC calculates model performance while accounting

for the increase in complexity associated with increased numbers of clusters. A description of the calculation of BIC is given by Pelleg and Moore [147], and is also presented in Appendix G. Briefly, it calculates the likelihood of the data under the model (which generally improves with increasing k) but applies a penalty based on the number of free parameters (which also increases with k). It is desired to select the BIC whose value is minimum, which represents the highest likelihood relative to the penalty.

In this approach, the target performance variables from the test module (e.g. adherence, power output) are used with BIC to determine an appropriate number of performance groupings. Figure 4-4 shows the BIC for a simple k-means algorithm run on all targets in this dataset from 1-20 clusters (using Levenshtein distance for adherence, a similar result is seen using PAM). As can be seen, on this dataset the BIC is a strictly increasing function. This implies that this dataset favors the use of as few clusters as possible, and the optimal (lowest) BIC indicates only a single cluster. This is an interesting finding, and indicates that the performances across the metrics in the dataset are similar enough that there are no natural groupings separating trainees. If it is hard to separate performances of trainees into groups, this may also indicate that supervised learning techniques may also encounter difficulties on this dataset (presented later in this chapter). However, while the BIC recommends the use of only a single cluster, this result has relatively little practical value for assessment. Grouping all trainees together provides little help in identifying groups of trainees that may be in need of training intervention (TI), as it only indicates the intervention should be applied to all trainees or none. However, this result does indicate that clustering approaches may have difficulty with this dataset, and may suggest the potential for future difficulties with classification approaches.

To find meaningful cluster assignments that are useful for training evaluation, other model selection techniques can be used. Figure 4-5 shows the sum of within-cluster distances for a simple k-means algorithm run on all targets (using Levenshtein distance for adherence, similar results for PAM). As the number of clusters increases, the overall distances between data points and their cluster centers will tend to decrease. However, the marginal gain for including an additional cluster will be low if data are already well fit by current clusters. Figure 4-5 demonstrates that the gain by adding additional clusters begins to fall off (the so-called "elbow method") at 3 clusters. Therefore, three clusters

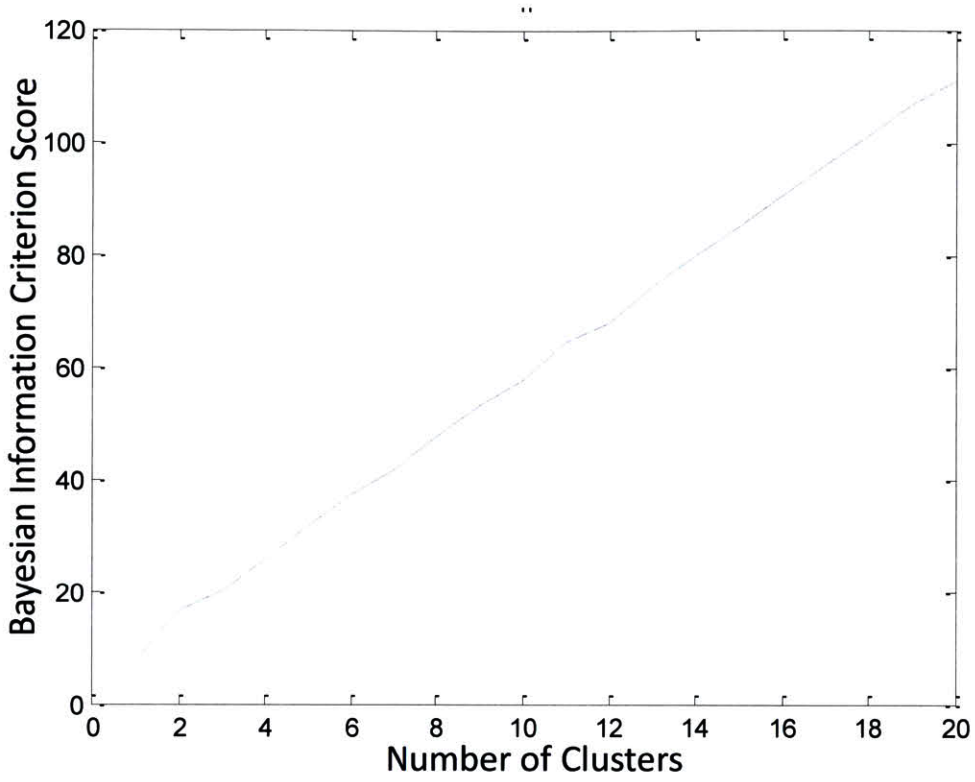


Figure 4-4: Selection of Appropriate Number of Clusters Through BIC.

are used in the creation of the models for the analysis in the following section. While trainees are not always consistent across all performance metrics, comparing these cluster assignments to the performance values can be interpreted as general performance categories. The cluster assignments indicated 8 trainees in the poorest performance category, 15 trainees in the middle performance category, and 23 trainees in the top performance category. Note that these assignments are over all of the test module performance variables, and thus represent clusters in 4-dimensional space.

To illustrate how these clusters identify performance categories, Figure 4-6 shows boxplots of the scores for each performance metric in the test module by clusters. The boxplots in Figure 4-6 represent the average performances for 30 runs of the k-means algorithm on the test module performance metrics. In the figure, it can be seen that Group 1 generally corresponds with the "poor" performers (having the lowest scores across all metrics), Group 2 with the "moderate" performers, and Group 3 with the "good" performers. While there is some variation in the groupings based on the particular

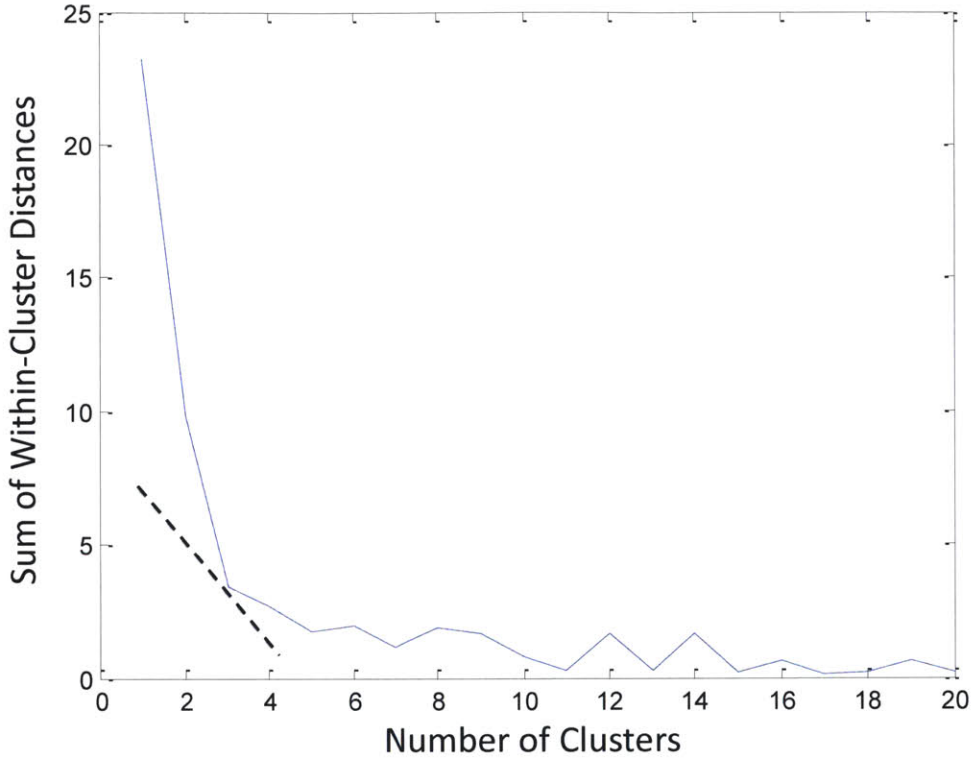


Figure 4-5: Selection of Appropriate Number of Clusters Through Elbow Method.

initialization of the k-means algorithm (represented by the variation in each boxplot), overall it is clear that these groupings could be used to generally divide trainees for targeted intervention techniques.

The clustering results on the full dataset (including metrics from the training modules) are presented in the following sections and are split into three investigations. The first is intended to identify which of the two adherence metrics (Levenshtein distance or PAM) provides better clustering performance on the nuclear reactor CBT dataset. Since cluster results are based on the ability to separate data in the feature space, this comparison can indicate the best adherence metric to utilize in later supervised analyses. For both Levenshtein and PAM, the targets used for external metric comparison were both the corresponding summative adherence metric and the power output in the test module. The second investigation considers the contribution of adherence, objective, and subjective information in cluster algorithm performance. The third investigation (presented in the dimensionality reduction section below), utilizes the ability of dimensionality re-

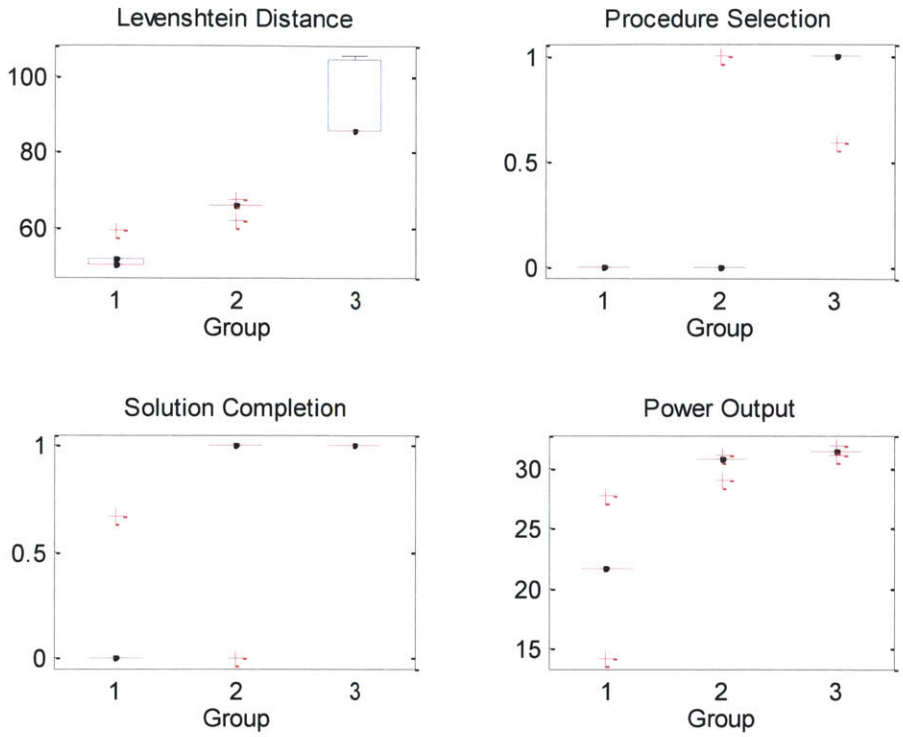


Figure 4-6: Boxplot of Performance Metrics in Test Module by Group. Median values are shown as central marks, edges of the box represent 25th-75th percentile. Whiskers are shown for data up to 1.5 times the interquartile range. Outliers are plotted individually with a "+".

duction to condense the action by action process-level information into a form that can be directly utilized by clustering techniques. In this investigation, the usefulness of the reduced process-level adherence features is compared to summative adherence features in cluster algorithm performance. The results of these investigations then contribute to the identification of which unsupervised algorithms have the best performance on this dataset, which metrics provide for the best clustering performance, and the usefulness of the process-level information provided by CBT environments.

Adherence Cluster Analysis

The first investigation compared clustering performance between models built on the Levenshtein distance and the PAM, to determine which metric provides better separability in the data given the 3 clusters determined by the elbow method. Table 4.2 shows the results

of this investigation, when using external metric comparisons with adherence and power output information. There are several important conclusions from these results. First, clustering techniques based on the Levenshtein distance tend to perform better than with the equivalent PAM models, due to their generally lower entropy and compactness scores. Therefore, it seems that the sequence-based Levenshtein distance may be preferable in unsupervised learning than the model-based PAM. Second, most of the better performing models across all metrics occur with simple clustering methods (k-means and hierarchical clustering). This indicates the more complex methods do not provide sufficiently better performance to justify their use, and it seems that on procedure-based datasets simple unsupervised algorithms are preferable.

The differences between k-means and agglomerative clustering can be teased out by looking at the resultant cluster assignments (an example set of assignments is shown in Appendix H). K-means tends to result in a more equal split between the clusters, which tends to result in greater compactness but low purity. Hierarchical clustering on this data tends to form a single large cluster, resulting in high purity and isolation (since the other clusters are used to fit outlying values). It is clear that for the power output external metrics, hierarchical clustering results in better performance. This is largely due to the nature of the power output variable; most trainees achieved a power output close to the maximum output, and only a few people received low power scores. Thus, hierarchical clustering which fits most trainees into a single cluster tends to perform well relative to this variable. However, from an evaluation perspective it is undesirable to have most trainees grouped into a single cluster. If most or all trainees are grouped together, there can be little to distinguish between trainees. From an intervention perspective, an intervention might be applied to a particular group; if most trainees are in the same group, the intervention would need to be applied to everyone. Thus, the more equal divisions provided by k-means are more useful for training evaluation, and thus k-means seems the optimal choice on this dataset. Therefore, in the comparison of adherence, objective, and subjective information in clustering performance, k-means is used as the underlying algorithm for comparison.

Table 4.2: Cluster Performance Comparison Between Adherence Metrics and Clustering Methods. Models with the best performance across clustering methods for each cluster performance metric are marked in gray.

Target	Features	Evaluation	K-means	Agglomerative Clustering	Gaussian Mixture Model	SOM
Adherence	Levenshtein Distance	Inspection	General Confusion	Favors single cluster	Data Incompatibility	Confusion between two clusters
		Entropy	1.23	1.52	Data Incompatibility	1.31
		Purity	0.33	0.28	Data Incompatibility	0.39
		Compactness	2.46	3.05	Data Incompatibility	2.66
		Isolation	2.36	6.41	Data Incompatibility	1.33
	PAM	Inspection	Cluster Ambiguity	Favors single cluster	Cluster Ambiguity	Cluster Ambiguity
		Entropy	1.33	1.54	1.37	1.37
		Purity	0.33	0.44	0.36	0.36
		Compactness	2.75	5.81	3.44	2.72
		Isolation	2.53	4.12	3.51	2.19
Power Output	Levenshtein Distance	Inspection	Cluster Ambiguity	Favors single cluster	Data Incompatibility	Cluster Ambiguity
		Entropy	1.12	1.02	Data Incompatibility	1.12
		Purity	0.33	0.7	Data Incompatibility	0.39
		Compactness	2.46	3.05	Data Incompatibility	2.66
		Isolation	2.36	6.41	Data Incompatibility	1.23
	PAM	Inspection	Cluster Ambiguity	Favors single cluster	Good overall performance	Cluster Ambiguity
		Entropy	1.13	1.02	1.08	1.15
		Purity	0.33	0.7	0.44	0.28
		Compactness	2.75	5.81	3.92	2.72
		Isolation	2.51	4.12	3.51	2.22

Clustering by Feature Type

In the second investigation, k-means was then used to investigate the contributions of different feature sets to cluster performance. This can indicate to training supervisors which metrics are the most informative for separating trainees into groups, which can help inform trainee intervention (TI) as discussed in Chapter 3. Specifically, features were divided into adherence (Levenshtein distance), objective (quiz scores and demographics), and subjective (model ratings and post-experiment survey) data types. Table 4.3 shows the cluster performance of each data type, using both adherence and reactor power output variables from the test module for the determination of "classes" in external metric calculations (entropy and purity). There is little difference in the external metrics between clustering with the adherence, objective, and subjective data types, as seen by looking across the entropy and purity values in Table 4.3. For internal metrics, it is clear that the adherence information provides better compactness, while the objective and subjective metrics provide greater isolation. This is partly a result of the higher dimensionality of the feature space for objective and subjective metrics. When downselected to the equivalent number of dimensions, the clustering algorithm returns empty clusters, indicating that the full dimensions are needed to fit the model. Thus, the adherence metrics appear to provide the greatest information per feature. However, there are considerably greater numbers of adherence features provided by the process-level information. The high dimensionality of these features prevent them from being used directly for clustering or supervised learning, and thus it is useful to turn to dimensionality reduction methods to trim down the feature set size.

Dimensionality Reduction

Dimensionality reduction (a form of feature selection) is another form of unsupervised learning that allows for the representation of data in high dimensional space in a lower-dimensional space. There are two primary reasons for this transformation: to remove redundant or irrelevant features and to try to avoid the curse of dimensionality. The first is fairly self-explanatory, where it allows the user to identify and remove features that are not providing useful content, which helps computation time and can prevent overfitting. The second relates to the idea of the curse of dimensionality, which states that as the number of dimensions increase, the separation between points approaches a

Table 4.3: Cluster Performance Comparison Between Adherence, Objective, and Subjective Metrics. Models with better performance are marked in gray.

Target	Evaluation	Adherence Metrics	Objective Metrics	Subjective Metrics
Adherence	Inspection	Slight general confusion	General Confusion	Favors single class
	Entropy	1.23	1.23	1.23
	Purity	0.33	0.33	0.33
	Compactness	2.47	40.6	33.18
	Isolation	2.31	8.36	8.23
Power Output	Inspection	Confusion between two clusters	General Confusion	Favors single class
	Entropy	1.12	1.13	1.12
	Purity	0.34	0.33	0.34
	Compactness	2.46	40.6	33.16
	Isolation	2.37	8.34	8.23

constant value. This creates great difficulties for algorithms that require measurements of distances or partitions of space, such as clustering algorithms and Support Vector Machines (SVMs).

The main strategy for dimension reduction revolves around feature extraction, which involves the transformation of data from high dimensional space into lower dimensions. The most common technique is Principal Component Analysis (PCA), which performs a linear mapping of the data into lower dimensional space. The objective of PCA is to obtain a smaller set of orthogonal projections along the original feature space in a way that the variance of the data along the new dimensions is maximized. By using the first several components generated, dimensionality reduction can be achieved (for further information on PCA, see [12]).

As previously mentioned, dimensionality reduction is particularly important on this dataset, due to the high number of features provided by the process-level adherence information. Dimensionality reduction can be used here for two different approaches: to assist with cluster analysis and to condense data prior to supervised analysis. The first approach utilizes dimensionality reduction to assist with cluster algorithm performance. As mentioned in the previous cluster analysis, the process-level adherence features could not be used directly due to the curse of dimensionality, and summative forms were used instead. However, it is possible that the summative adherence information does not

capture useful information contained within the process-level data. Instead, it is possible to use dimensionality reduction techniques instead as an alternate way to compress the adherence information. Thus, clustering could be performed again on a set of PCA reduced features, to see if the information contained in the process-level information provides benefits to cluster algorithm performance.

Cluster Analysis on PCA-reduced Features

A third clustering investigation was conducted utilizing the PCA-reduced features. To identify any benefits of dimensionality reduction in clustering on this dataset, the adherence, objective, and subjective metrics were each processed using PCA, and the first three principal components were used to repeat the cluster analysis comparison shown in Table 4.3. The results are presented below in Table 4.4. It can be seen that as expected the compactness and isolation ratings decreased for the objective and subjective metrics, due to the reduced dimensionality of the feature space. However, it is clear that the adherence metrics still achieve better compactness, while the other metrics provide better isolation performance. There are three major conclusions from these results. First, the PCA-reduced adherence performance (shown in Table 4.4) was identical to that of summative adherence (shown in Table 4.3) for both external and internal metrics. This indicates that there is no advantage to the use of the action-based adherence information compared to the module-level summative adherence information in the cluster analysis. Second, all three metric types show similar performance in clustering approaches, and thus it may be useful to collect all these metric types when clustering procedural training data. Second, the performance of the clustering algorithms by inspection (compared to "classes" based on test module performance) was generally relatively poor. So while clustering may be able to find groups of similar trainees based on training features, these do not appear to provide a good match to test module performance. This indicates that there may be difficulties with relating training module performance to test module performance, which is investigated in supervised learning later in this chapter.

Dimensionality Reduction for Supervised Approaches

The second dimensionality reduction approach relates to the reduction of the number of features prior to supervised learning to prevent overfitting. Since the primary contributor

Table 4.4: Cluster Performance Comparison Between Adherence, Objective, and Subjective Metrics, Normalized to Three Features. Models with better performance are marked in gray.

Target	Evaluation	Adherence Metrics	Objective Metrics	Subjective Metrics
Adherence	Inspection	General confusion	General Confusion	General Confusion
	Entropy	1.23	1.22	1.23
	Purity	0.33	0.33	0.34
	Compactness	2.47	18.81	13.82
	Isolation	2.31	7.77	7.94
Power Output	Inspection	Confusion between two clusters	General Confusion	General Confusion
	Entropy	1.12	1.13	1.13
	Purity	0.34	0.32	0.34
	Compactness	2.46	18.96	13.81
	Isolation	2.37	7.75	7.95

to feature space dimensionality on this dataset is the process-level information, PCA was applied to the process-level Levenshtein distance features to create new features for supervised learning. While the intent is to reduce the number of features prior to use in supervised learning, there is little obvious guidance as to how many features should be used. However, general guidelines can be drawn from the discussion of overfitting in Chapter 3. Specifically, it is desired for the number of features to be much less than the number of data points. Given 47 trainees in this dataset, it would be undesirable for the number of features to exceed 10-20. Thus, to be able to combine these process-level adherence features with other objective and subjective features, it is preferable to reduce the number of these features to less than 10. Given that there are three modules, it seems reasonable to allow for three features each to be derived from PCA on the process-level adherence metrics. Using this strategy, new features were generated based on the first three principal components from each module. The details of these three principal components are given in Appendix I. These features are identical to the PCA-reduced adherence features used in cluster analysis above, but are considered specifically as features in supervised learning. In the later sections on supervised learning, these metrics are referred to as "process-level adherence" in the analysis.

4.4.2 Summary of Unsupervised Analysis

This section presented the results of two unsupervised learning approaches on the rule-based dataset: clustering and dimensionality reduction. Clustering can find groupings amongst the trainees, which could assist training evaluators in selecting groups of trainees for intervention. On this dataset, the BIC indicated the use of a single performance cluster, which may indicate that there may not be clear performance groups that emerge in the data. The Elbow method discussed here here was used as an alternative method for selecting an appropriate number of clusters (in this case three clusters), which provided a more meaningful cluster distribution for training evaluators. Specifically, these cluster assignments were able to generally separate trainees into "poor", "moderate", and "good" overall performance categories. Since each group had similar performance, any intervention needed could be applied to an entire performance category.

The results from the series of clustering investigations provided insights into the selection of particular parameters and features for clustering on this dataset. Specifically, the first investigation using the elbow method indicated the use of three clusters. The results from the second investigation (shown in Table 4.2) indicated that simple clustering methods have equivalent or better performance to more complex methods, strongly suggesting the use of simple algorithms on similar rule-based training datasets. Additionally, the clustering results indicated that the Levenshtein distance provided equivalent cluster performance to the PAM, indicating that it may be preferred computationally in machine learning approaches. The third investigation demonstrated that all three feature types (adherence, objective, and subjective) were equally useful to cluster algorithm performance. The final investigation demonstrated that the module-level summative adherence metrics were more useful for clustering than the action-level metrics.

Based on these results, we can suggest an "ideal" set of features from the dataset to be used in unsupervised analysis. In particular, this model would include the summative Levenshtein adherence metrics, the objective metrics (quiz scores and demographics), and the subjective metrics (module ratings and post-experiment survey). Using these metrics, the trainees can be clustered into three groups. As previously mentioned, these groups could serve as indicators for group TI. However, the question arises of how well groups based on these metrics (from the training modules) are good separators of test module performance. To test this, this "ideal" model was used to separate trainees on this datset

into three groups. A set of boxplots (each representing 30 runs of k-means) using these splits was created similar to Figure 4-6 on the test module performance metrics, and is shown in Figure 4-7. While the separations between groups are not as clear as in Figure 4-6 (which was based on the final performance metrics directly), it can still be seen that the training metrics can still be used to create clusters that still have meaning to training supervisors as indicative of performance categories. These groups could then be used by the supervisor to inform training intervention techniques.

While the clustering methods were able to separate the data into three groups, the algorithms do not specify meaning to these groups. Any interpretation of a cluster as "good" or "poor" performers would have to be made by a training evaluator, such as the interpretation provided above based on analyzing the cluster assignments with respect to the performance scores. Additionally, since the BIC on the final performance metrics

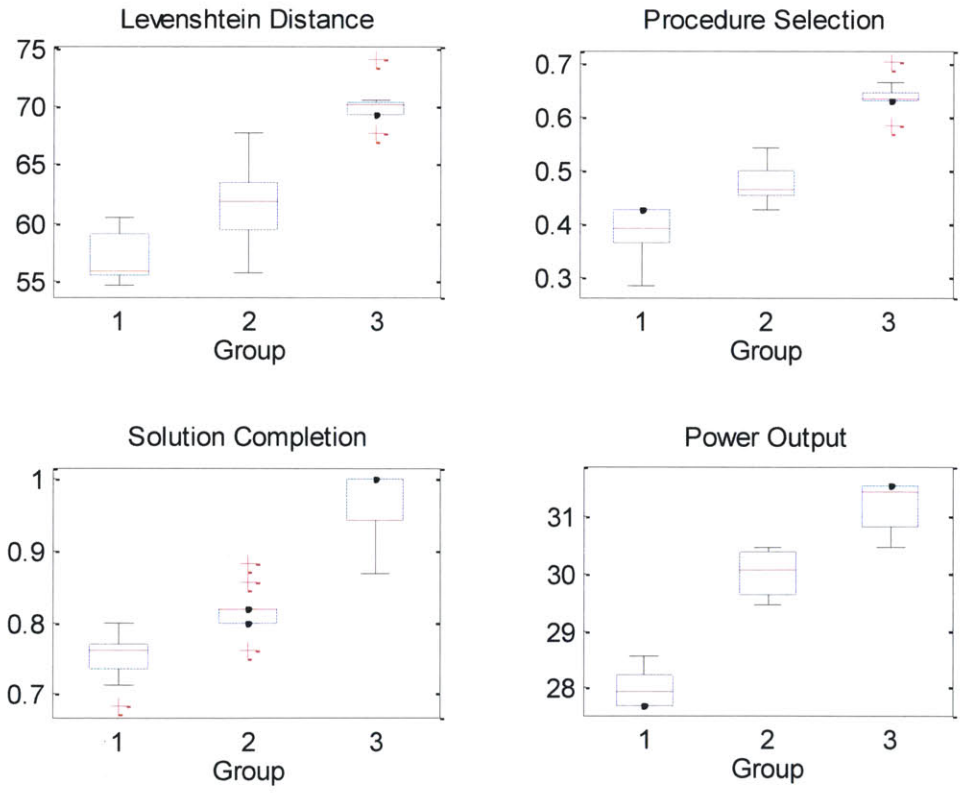


Figure 4-7: [Boxplot of Performance Metrics in Test Module by Group using "Ideal" model. Median values are shown as central marks, edges of the box represent 25th-75th percentile. Whiskers are shown for data up to 1.5 times the interquartile range. Outliers are plotted individually with a "+".

indicated the presence of only a single cluster, it is possible that the clusters of trainees may contain differences in behavior during training modules, but little difference in test module performance. This presents an important lesson from this dataset: while clustering algorithms will generate a result given any dataset, clusters on complex training datasets such as the rule-based dataset presented here may not have useful interpretations for a training evaluator.

Dimensionality reduction was also applied to this dataset through the use of PCA, which was particularly useful for reducing the number of features associated with process-level procedure adherence metrics. It was demonstrated that PCA could be utilized to condense the process-level features prior to use in supervised learning models. Without dimensionality reduction, these features could result in overfitting in supervised learning approaches. The methods and results for supervised learning on the rule-based dataset utilizing the PCA-reduced features are presented in the following section.

4.5 Supervised Learning

As discussed in Chapter 2, supervised learning utilizes labeled data to form relationships between features and labels in the data. If the labels represent a state at a future time compared to the features, this can be interpreted as a predictive approach. In training, supervised learning can use in-training features to predict later-training or post-training performance. This is useful in several applications as discussed in Chapter 3, particularly for informing TI by providing early feedback on trainees that are predicted to have poor performance. Supervised learning can include both regression and classification techniques, dependent upon the nature of the target variable (continuous and discrete, respectively). As identified in Table 4.1, there were both continuous targets (adherence and power output) and discrete targets (correct procedure selection and solution completion) in this rule-based dataset. Therefore, it is appropriate to utilize both regression and classification techniques on this dataset to analyze prediction of all the target variables.

There were several main goals of this analysis. The first was to identify which supervised machine learning algorithms resulted in the best prediction performance, as tested by post-hoc analysis. The second goal focuses on temporal prediction, or the improvement in prediction over time as the trainees move through the training program. This

second analysis is able to inform the usefulness of these methods in informing training intervention. The following sections present the methods and results of the supervised learning algorithms relative to each of these goals.

4.5.1 Supervised Algorithms

A range of supervised learning algorithms were introduced in Chapter 2 that cover the most common types of algorithms. As a reminder, these are linear regression, logistic regression, support vector machines (SVMs), and artificial neural networks (ANNs), listed roughly in order of increasing complexity. Linear regression and ANNs are commonly used for regression problems, where the labels are measured on a continuous scale. ANNs are also well suited for classifications approaches, as are logistic regression and SVMs. In a classification problem, the labels are discrete. Each of these algorithms were investigated for use on the knowledge-based dataset, in both regression and classification approaches as appropriate.

In this rule-based dataset, there are two continuous targets that represent final training outcomes: Levenshtein distance (adherence) and power output in the test module. In the analysis below, two regression methods were tested on the these targets: linear regression and artificial neural networks (ANNs). For the discrete targets (correct procedure selection and solution completion), classification methods used included logistic regression, SVMs, and ANNs. For both regression and classification, generalization performance was assessed using a Leave-One-Out (LOO) strategy, where a single data point is held out from each model as a test point, and the remaining data points are used for training and validation of the model. For ANN models, which require validation data to adapt the model weights, the remaining data (without the single test point for each model) was split with 80% training, 20% validation. For all other model types, all of the remaining data points were used in the training of the model.

Both SVMs and ANNs also have model parameters that must be selected during the creation of these models. In SVM models, the choice of kernel (such as linear, polynomial, or Gaussian) can impact the capability to fit functions. In this analysis, the standard linear kernel was implemented to maximize the simplicity of the resultant models. For determining the structure of the ANN models, several heuristics can be used. For nearly all mappings of the input space into the output space, one hidden layer is sufficient

to characterize the mapping [148] and therefore was selected to be used for the ANNs created in this study. The optimal number of neurons in the hidden layer depends on several variables including the size of the training set, the complexity of the underlying function, the noise, and the activation function selected [149]. However, general heuristics have suggested that the number of training data should be between 2 and 10 times the sum of the size of the input, hidden, and output layers (e.g. [150]). In this study, given the number of data available and the general range of the input layer, a moderate network size of 10 hidden neurons was selected.

In both regression and classification approaches, two prediction strategies were analyzed: post-hoc prediction accuracy and temporal prediction accuracy. Post-hoc prediction utilizes all of the data available throughout the entire training program, and attempts to achieve the highest accuracy relationships between the features from the training modules and the targets from the test module. This strategy can identify the prediction algorithms that result in the best prediction performance across the available feature sets. In this way, the analysis can suggest the appropriate supervised learning algorithm to use on similar rule-based training datasets or further analyses, such as the temporal analysis presented later in this chapter. It may also be useful from a quality assurance perspective to training designers by identifying which features provide the greatest contribution to prediction accuracy. By understanding which metrics are useful for prediction, the designer can adjust the program to focus on these elements and/or remove assessment metrics that provide little predictive value.

An analysis of temporal prediction accuracy compares the prediction accuracy of models created as the trainee moves through the training program. In a realistic prediction setting, the training evaluator would like to have an accurate prediction of a trainee's performance early in the training program, when only some of the features are available for the model. To investigate the development of prediction accuracy over time, models can be created after each training module, incorporating only the information from that module and prior modules. This approach can provide a sense of how these prediction models can be used to inform TI. If accurate predictions of future trainee performance are available early in a training program, the evaluator can apply intervention techniques to improve learning when needed. In particular, CBT environments able the collection of more frequent process-level information (such as the adherence information in this

dataset), and it is of interest to determine whether these elements are able to contribute to TI over traditional summative metrics. In the following sections on regression and classification, both post-hoc and temporal analyses are conducted to test the usefulness of supervised learning algorithms in training evaluation settings.

4.5.2 Regression Results

As previously discussed, the target variables used in regression approaches were the Levenshtein distance and the power output in the test module. This section presents both the results of a post-hoc analysis to analyze the prediction accuracy of each method, as well as a temporal analysis of the benefits of process-level information in the regression setting.

Post-hoc Prediction Results

The selected regression methods (linear regression and ANN regression) were compared based on the model fit of the data on the test set (generalization performance), as measured by Mean Squared Error (MSE) between the predicted values and the true values. Table 4.5 shows the results of the post-hoc regression analysis. In this table, "Summative Adherence" refers to the Levenshtein distance at the end of each module, while "Process-Level Adherence" refers to the PCA-reduced set of features generated from the process-level (action-by-action) adherence measurements. As a reminder, both of these metrics are generated using action-level data, but in summative form only the module-end value is used while for the process-level form the values at each action are included as features. "Total Summative" refers to the combined use of objective, subjective, and summative (end of module) adherence metrics as features, while "Total Process-level" refers to the use of objective, subjective, and PCA-reduced adherence metrics. Note that the scale of MSE is much larger for the models predicting adherence than those predicting power output. This relates to the scale of the target variables, which ranged from 0-168 for adherence and 0-32 for power output on this dataset. Thus, it is anticipated that errors will be larger for adherence, and correspond to generally higher MSE.

In Table 4.5, simple linear regression models show better performance (as measured by MSE) than the equivalent ANN models on all feature sets except the total process-level feature set. This result is found for both adherence and power output prediction targets.

This indicates that despite the complexity of this rule-based dataset, the simple models are able to capture the relationships between features and targets as well as the more complex models. While the ANN models perform better than linear regression when using the total process-level feature set, it can be seen that the MSE for these models is higher than other models for the same target, indicating poorer generalization performance. For linear regression, a similar finding is shown by the other combined feature sets (objective + subjective and total summative), which generally have higher MSE than models using adherence, objective, and subjective feature sets individually. A possible explanation for the poorer performance of the linear regression models with the larger feature sets comes from overfitting. By utilizing a larger number of features, the linear regression models may be fitting the noise in the training data, which results in poorer performance on previously unseen data. The ANN approach does not show this same trend, which could be due to the greater flexibility of ANNs for larger feature sets, or could also be impacted by the use of separate validation data to adapt the model weights. However, since linear regression resulted in the models with the best overall generalization performance for both targets, and generally better MSE than equivalent ANN models, linear regression is selected as the algorithm for use in the temporal analysis presented later in this section.

A comparison between the model performance shown in Table 4.5 and the ranges of the target variables of interest can give an idea of the overall usefulness of the models. By selecting the best performing models for both targets (linear regression using subjective features), the MSE is shown to be 1583 for adherence and 41.84 for power output. By taking the square root, these can be put back into the units of the original variables, and indicate that the adherence predictions were off by an average of 39.8 for adherence and 6.5 for power output. While these are much smaller than the ranges observed for the respective variables (0-168 for adherence, 0-32 for power output), it represents a relatively high error relative to the distributions of the target variables across trainees (standard deviation 33.5 and 5.5 for adherence and power output, respectively). Thus, the average model error is greater than one standard deviation of the data, and this indicates that the model would have difficulty separating trainees with similar performance. This indicates that for the rule-based dataset, there is general difficulty in relating training performance in modules 1-3 to adherence and reactor power output during the final test module.

The information contained in Table 4.5 can also provide indications of the contribu-

Table 4.5: Post-hoc Regression Results. Models with better performance between the two algorithms are marked in gray.

Target	Feature Set	Linear Regression MSE	ANN Regression MSE
Adherence	Summative Adherence	1112 +/- 1391	4631 +/- 1374
	Process-Level Adherence	3168 +/- 3819	11868 +/- 4090
	Objective	1785 +/- 712	12368 +/- 2274
	Subjective	1583 +/- 734	6008 +/- 1221
	Objective + Subjective	5040 +/- 1834	7339 +/- 1264
	Total Summative	4943 +/- 1615	6681 +/- 1279
	Total Process-Level	24164 +/- 10996	8551 +/- 1510
Power Output	Summative Adherence	29.54 +/- 44.04	121.68 +/- 38.67
	Process-Level Adherence	181.45 +/- 300.17	289.53 +/- 127.41
	Objective	49.95 +/- 25.82	312.95 +/- 77.42
	Subjective	41.84 +/- 26.84	178.78 +/- 34.12
	Objective + Subjective	112.69 +/- 32.20	264.34 +/- 49.16
	Total Summative	99.69 +/- 25.16	187.51 +/- 33.04
	Total Process-Level	625.03 +/- 490.41	209.85 +/- 41.31

tions of each feature type to prediction accuracy. Of particular interest are the contribution of adherence information, the collection of which is enabled by the use of CBT technologies. In looking at the individual feature sets in Table 4.5 (first 4 rows), it appears that the highest prediction accuracy is provided by summative adherence metrics, followed by subjective metrics, while objective metrics provide the least prediction accuracy when used alone. The contribution of adherence information can be further noted by comparing the models with and without adherence information. By comparing the last three rows for each target in Table 4.5, we can see that the models that include summative adherence information perform better than the models using only objective and subjective information, while process-level adherence features appear to have much worse performance. For example, the linear regression MSE for models without adherence information (5040 and 112.69 for adherence and power output targets, respectively) are higher than those that include summative adherence information (4943 and 99.69) while much lower than those with process-level adherence information (24164 and 625.03). This indicates that summative adherence metrics contribute to post-hoc prediction accu-

racy beyond the objective and subjective metrics alone, while the action-level adherence metrics are not predictive of later performance. Since the performance of the individual metric models are generally better than the combined models, the individual summative adherence models result in the best prediction performance on this dataset. The top three feature sets that result in the best prediction performance for each algorithm are summarized in Table 4.6.

An important application of supervised learning to training datasets includes the prediction of future trainee performance. While the post-hoc analysis presented above can help to identify the features that are most informative once the program is completed, to assist with TI predictions must be made prior to the end of the training program. Thus, it is worthwhile to investigate the development of prediction accuracy over the course of the training program to determine the usefulness of supervised approaches in informing the timing of TI.

Temporal Prediction Results

A second investigation was conducted to explore the development of model fits over time using linear regression to fit power output in the test module as a target variable. As the trainee moves through the training program, more information becomes available to an evaluator. Thus, datasets were created that were representative of information available after each module. For example, after Module 1, the module 1 adherence information, module 1 quiz score, module 1 subjective ratings, and demographic information would

Table 4.6: Feature Sets with Best Regression Performance. Targets are performance metrics from the test module, while the feature sets listed are drawn from the training modules.

Algorithm	Target	#1	#2	#3
Linear Regression	Adherence	Summative Adherence	Subjective	Objective
	Power Output	Summative Adherence	Subjective	Objective
ANN Regression	Adherence	Summative Adherence	Subjective	Total Summative
	Power Output	Summative Adherence	Subjective	Total Summative

already be available. However, the equivalent information for later modules would not yet be available. Figure 4-8 shows model generalization performance (using the same LOO strategy) at the end of each module using only the information available both with and without the process-level adherence information.

There are several important findings from Figure 4-8. First, the model error actually increases as more information is incorporated into the model construction. This is an interesting finding on this dataset, as it indicates that the best time to make predictions of future performance is after the first module. This seems counterintuitive, but the result follows one of the most important challenges that was identified in Chapter 3: overfitting. As more features are included in the model, they may be used to fit noise within the training data rather than the underlying relationships. The increasing error shown in Figure 4-8 indicates that the extra features from additional modules are contributing to overfitting, and create sensitive models that have trouble predicting previously unseen

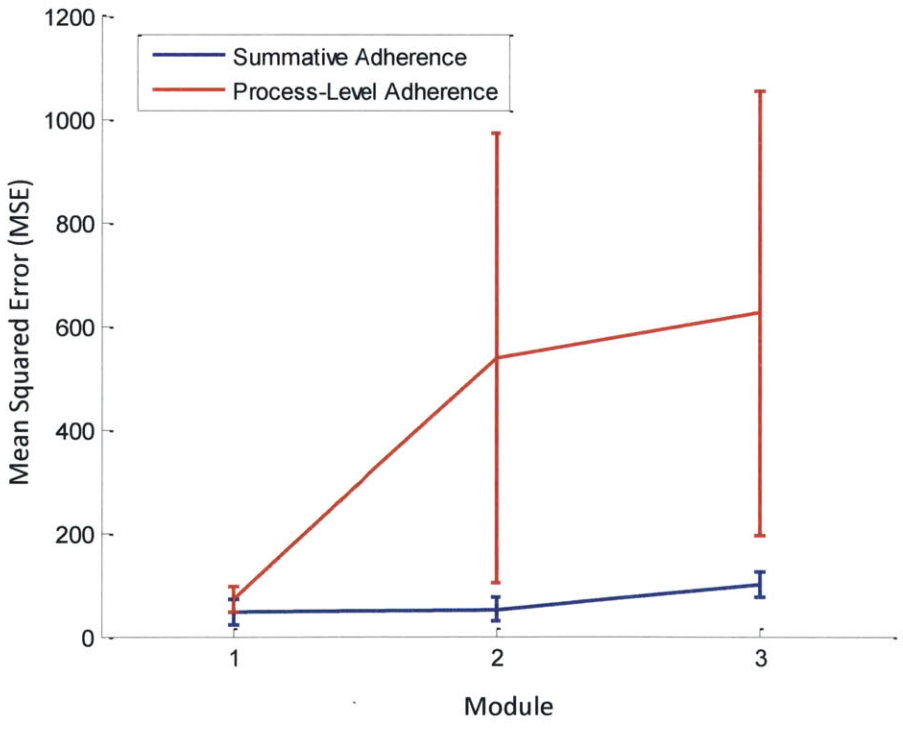


Figure 4-8: Prediction Accuracy of Linear Regression Models By Module. Data shown is mean +/- standard error.

data. Therefore, Figure 4-8 does not necessarily suggest that the features in Module 2 and Module 3 are not useful at all, but rather that through overfitting the increased number of features creates difficulties in generalization performance. This result does create difficulties in utilizing this method for recommending intervention timing for training evaluators, and rather suggests that the evaluator use fewer rather than more features when predicting performance.

Second, similar to the post-hoc analysis, the summative adherence models have better performance than those using the process-level adherence information for all three modules. Third, at Module 2 the process-level model error increases dramatically as compared to the summative model, and shows very wide error bars. This indicates that the model accuracy varies widely from trainee to trainee, and a more detailed explanation can be found by looking at the model predictions. In Module 1 (the simplest module), all trainees were able to generally follow the procedures, and thus generated action-level data that was similar to each other. In this case, the model is making predictions for previously unseen data that has feature values in the same range as those used for training the model. In Module 2, a few trainees became very lost early on in the procedures, and began performing actions seemingly randomly with the interface. The resultant adherence behavior of these trainees was markedly different than the rest of the trainees. Thus, when the model tries to predict the performance of one of the poorly-performing trainees, it struggles to use the same simple relationships used to train the model on feature values that are well outside the expected range.

However, this raises the question of why the summative adherence information did not have this same difficulty? A probable explanation lies in the nature of each feature. At the summative level, any trainees that get lost will have similarly poor adherence scores, and can be used to help make predictions on other struggling trainees. In the process-level information, trainees who get lost may have radically different feature values dependent upon *when* they became lost in the procedure. For example, if one trainee gets lost at step 1, s/he will have a very poor adherence for step 1. If another trainee becomes lost at step 4, s/he will similarly have a poor adherence value for step 4. When the process-level information is included in the model training, it treats these cases differently (e.g. "Trainee A became lost at step 1"). At the summative level, where the trainees have similarly poor adherence scores, this information essentially tells the model that "the

trainee became lost."

This result provides an important lesson for training evaluators attempting to use machine learning on training data. The models created assume that the behaviors of new trainees will be similar to those of prior trainees (used in the training of the model). If features are selected that make trainees unique from each other, as with the process-level information here, the generalization performance of the model will be poor. In Figure 4-8, it is clear that the summative adherence information is better able to avoid this issue of features that make some trainees different from all other trainees, maintaining the generalizability of the model.

4.5.3 Classification Results

As previously discussed, there are several discrete targets on this dataset (correct procedure selection and solution completion) that require the use of classification approaches for supervised learning. Thus, the regression analysis was repeated using these categorical target variables and classification techniques: logistic regression, SVMs, and ANNs. The primary performance variable of interest in this analysis was the classification error rate. Table 4.7 shows the post-hoc results for each classification technique. There are several important conclusions from these results. First, ANNs seem to result in generally the best performance across the three algorithms tested, primarily using the largest feature sets (total summative and total process-level). This may be due to the greater flexibility of the algorithm to fit non-linearities in the more complex feature sets. Second, logistic regression runs into computational difficulties when utilizing the combined datasets, while SVMs and ANNs do not. This issue relates to the rigidity and assumptions made by logistic regression compared to the more flexible SVM and ANN methods. Overall, on complex datasets such as from the procedure-based environment presented here, it appears that the flexibility of ANN algorithms is useful in classification tasks. Third, it is apparent that while the combined feature sets generally offer the best performance, reasonable performance is able to be achieved though each of the individual feature sets (adherence, objective, and subjective). This indicates that all of the metric types used may contribute to model fits in classification approaches.

Last, there are major differences between the prediction accuracy of the various target metrics. Procedure selection generally has the poorest prediction accuracies. The lowest

Table 4.7: Post-hoc Classification Results. Data indicate classification error (%) using each feature set to predict each target listed. Best performing methods across algorithms are shown in gray.

Feature Set	Target	Logistic Regression	SVM	ANN
Summative Adherence	Procedure Selection	43.2 +/- 0.126	46.0 +/- 1.43	44.44 +/- 1.656
	Solution Completion	17.3 +/- 0.108	20.0 +/- 1.33	22.22 +/- 1.171
Process-Level Adherence	Procedure Selection	51.4 +/- 0.132	43.0 +/- 1.160	53.328 +/- 1.147
	Solution Completion	22.2 +/- 0.128	28.0 +/- 1.033	21.109 +/- 0.820
Objective	Procedure Selection	38.0 +/- 1.229	52.0 +/- 1.549	54.439 +/- 2.694
	Solution Completion	23.0 +/- 1.767	17.0 +/- 1.160	11.11 +/- 1.047
Subjective	Procedure Selection	43.0 +/- 1.703	34.0 +/- 1.578	35.552 +/- 1.366
	Solution Completion	28.0 +/- 1.932	21.0 +/- 0.876	16.665 +/- 1.309
Total Summative	Procedure Selection	Data Incompatibility	49.0 +/- 0.994	52.217 +/- 1.576
	Solution Completion	Data Incompatibility	14.0 +/- 0.843	17.776 +/- 1.500
Total Process-Level	Procedure Selection	Data Incompatibility	52.0 +/- 1.751	42.218 +/- 1.366
	Solution Completion	Data Incompatibility	18.0 +/- 0.789	15.554 +/- 1.405

error rates are achieved on predicting solution completion (15-25%). However, it is informative to compare these accuracies to a more "naive" approach. For example, a natural strategy might be to classify all trainees into the most common class in the variable. For the variables presented, this would result in an overall error rate of 48.9%, and 17.0% for procedure selection and solution completion, respectively. When comparing the outputs of the algorithms to these "naive" models, it is clear that there is little to no improvement from the machine learning methods. Thus, even though the results indicate that ANNs performed the best of the three classification methods tested, the classification approach overall does not yield meaningful results over even simpler percentage based models. Since the post-hoc models (using all information available from training) do not improve predictions over naive approaches, models based on subsets of these features

in a temporal analysis demonstrate similar poor prediction performance. Thus for this dataset, the classification approach was not able to aid evaluators in informing TI timing over simple naive models.

4.5.4 Discussion of Prediction Results

The results indicate several important findings for the use of machine learning algorithms in this rule-based training setting. Post-hoc regression results revealed the importance of summative adherence information for post-hoc prediction. As seen in Tables 4.5 and 4.6, the models using summative adherence alone resulted in the best overall generalization performance across all the feature sets for both methods. This result was not found for the models with process-level adherence information, which generally had poorer prediction performance than all other feature sets. Given that both the summative and process-level adherence information are fundamentally generated in a similar way, it is interesting that one version of adherence would result in the best performance while another the worst. There are several possible explanations for this result. First, the process-level information could not be included in the models in its raw form, due to the high number of features it contains. It is possible that the PCA process to reduce the dimensionality of the process-level information removed a considerable amount of useful information for prediction. Second, the process-level metrics measure performance for individual actions or mistakes on steps by the trainees, while the target performance metrics represent an overall measure of the trainee performance on the final module. Thus, there may be little relationship between individual mistakes in the training modules and patterns of overall performance on the test module. Using the same logic, it seems reasonable that overall (summative) measurements of adherence during training would correlate well with adherence behavior during the test module. This is important from a training evaluation standpoint, as it indicates that adherence (an important part of safety in rule-based environments) as well as other performance metrics are consistent with adherence to procedures during the training process itself.

For all of the methods and targets in Table 4.6, using summative adherence rather than the process-level adherence improved prediction performance. However, as a reminder the calculation of the module-level "summative" adherence still requires the measurement of adherence for each action, as described in Equation C.1. This indicates that

the action-level adherence information is useful for trainee performance prediction, but only when modeled using an overall, module-level form. Since the summative form of adherence provided the lowest generalization prediction error rates, process-level adherence information should be collected as part of the assessment regimen, but converted to summative form prior to use as features in supervised learning. Additionally, the post-hoc regression analysis showed that the simpler linear regression algorithm out-performed the more complex ANN algorithm every feature set and target except for the prediction of adherence in the final module based on summative adherence from the training modules (see Table 4.5). This finding indicates that on similar rule-based datasets, simple linear regression may be preferred to ANN regression.

A temporal analysis of prediction performance over time indicated similar results as the post-hoc analysis, showing that the summative adherence information was preferable to the process-level information for generalization performance. Specifically, it was found that the nature of the process-level information created uniqueness across trainees that created difficulties in prediction accuracy for a few struggling trainees. Additionally, it was clear from the temporal analysis that overfitting is a major issue on this rule-based dataset, and using fewer features improved generalization performance. Due to the overfitting issues, the ability of this method to make recommendations for the timing of TI is limited, as it merely suggested that the most accurate predictions could be made at Module 1 due to the fewest features being used at that time.

The classification analysis provided different results from regression in the selection of machine learning algorithms. In Table 4.7 it is seen that the simplest method (logistic regression) encountered numerical difficulties when using the larger feature sets, while the more complex SVM and ANN models did not have similar difficulties. While it would be possible to reformat the data to remove these errors, the purpose of this analysis is to investigate these algorithms on a typical dataset as would be used by a training evaluator. In its basic form, the dataset is not compatible with the logistic regression algorithm, which indicates a potential weakness of the algorithm on similar rule-based datasets. Rather, the results presented in Table 4.7 indicate that the ANN approach provides the lowest classification error and therefore best performance on this dataset.

However, there is an important caveat to the classification findings. The performance of the models for all feature sets were not noticeably better than the naive error rates of

48% and 17% that are obtained by assigning a single class to all trainees. This indicates that the classification approach was not appropriate for this dataset, and could be the result of the nature of the targets (procedure selection and solution completion), or could be related to the features themselves. In particular, a lesson may be drawn from the unsupervised analysis presented earlier in this chapter. The BIC indicated the presence of only a single cluster based on the features in this dataset. It is possible that this mirrors the classification results, indicating that in this dataset there do not exist strong predictive relationships to separate trainees into categories. It also sends an important lesson to training evaluators using these predictive methods on similar datasets: the model will output a prediction value for any dataset, but this does not guarantee that the resultant model is useful for prediction. Since the classification models cannot provide better prediction than the naive models, it would be inappropriate to use these models to drive quality assurance or TI.

4.6 Summary

This chapter presented the selection, methods, and results of both unsupervised and supervised approaches on a rule-based dataset. The findings from this analysis has provided insights into the usefulness of these methods for assisting with training evaluation in rule-based environments. The main takeaways from this work include:

1. The performance of the simpler clustering algorithms (k-means and hierarchical clustering) in external and internal metrics indicate that these methods are preferred to more complex GMM and SOM clustering approaches on this rule-based dataset. All three types of metrics (adherence, objective, subjective) contributed to cluster algorithm performance.

2. The BIC analysis indicated that this dataset may be best represented using a single cluster, and splitting trainees into categories based on performance may be difficult on datasets similar to the one presented here. This finding was reflected in the results of the supervised classification analysis, which indicated that the most favorable model performances was equivalent to predicting all trainees into a single performance category.

3. An alternative strategy using the "elbow" method identified three clusters, which provides a more useful split for training evaluators. Cluster algorithm results using three clusters were able to divide trainees into "poor", "moderate", and "good" performance categories. A series of analyses indicated that cluster performance was best on this dataset when using summative Levenshtein distance along with objective and subjective metrics.

4. Dimensionality reduction was critical on this datset to allow for the use of process-level adherence metrics in supervised analysis. Similar datasets that collect action-by-action information will likely need to use similar approaches to reduce the dimensionality of the feature space to prevent overfitting.

5. Summative adherence metrics from the training modules provided the best overall performance in prediction of adherence and power output in the test module. This indicates that adherence measurement as described in this data collection experiment is predictive of future procedure adherence and overall performance.

6. Process-level adherence information did not improve prediction over summative adherence or other feature types (subjective, objective). While the information may be useful in summative form, the inclusion of action-by-action adherence information is not useful for prediction models or descriptive clustering approaches.

7. Regression results suggested that the simpler linear regression approach was preferred to the more complex ANN models. However, as shown in the classification analysis, the assumptions in the simpler models can create numerical difficulties on complex datasets such as the rule-based dataset presented here.

There are several important limitations of this analysis based on the properties of the dataset used. First, the number of trainees was very limited, which likely played a major role in the issues related to overfitting in supervised approaches. Second, the collected dataset was limited based on logistical considerations, and the training program only utilized three training modules followed by a test module. The shortened training program as compared to typical rule-based environments may have limited the ability to see temporal benefits of process-level information. Third, the trainees used in the data collection process described in this chapter were generally novices to nuclear power

plant environments. Thus, these results may not generalize well to retraining of veteran operators.

This chapter has presented findings from machine learning approaches on a rule-based training setting: a procedure-based training environment. The procedure-oriented task structure, high numbers of process-level features and the selection, and use of adherence metrics in this environment impacted the results of machine learning approaches on this dataset, which showed mixed success. This raises questions about how such approaches will generalize to knowledge-based environments, or if the changes in training structure and metrics will alter the results of applying machine learning approaches to a knowledge-based dataset. To answer these questions the next chapter, Chapter 5, presents a similar analysis to this chapter, utilizing data from a representative knowledge-based training setting: a classroom training environment.

Chapter 5

Application of Machine Learning to Knowledge-Based Training Environments

As discussed in Chapter 1, one of the important types of training environments to consider for trainee assessment is structured around knowledge-based tasks. These environments commonly focus on a traditional classroom style of training, utilizing summative metrics for assessment such as projects, presentations, or examinations. Chapter 2 introduced the changing landscape of knowledge-based training due to the rising popularity of online and computer-based training methods. The larger, process-level data sets that can be obtained from these newer systems may have assessment benefits in addition to the traditional summative information (e.g. examinations) in terms of accuracy, consistency, or timing. To determine the usefulness of machine learning algorithms in knowledge-based settings for assessment both with and without process-level information, an exemplary classroom dataset was obtained that incorporates both traditional classroom elements as well as online components. Both unsupervised and supervised machine learning algorithms were then applied to this dataset to investigate the applicability of the algorithms.

This chapter is divided into three main sections. The first section provides a brief overview of the data collection methods and the dataset used in the analysis. The second section outlines the unsupervised learning approaches and results on the dataset. The third section reports the equivalent methods and results from the supervised approaches. Within each of the latter two sections, a set of conclusions and recommendations is

presented based on the results for each machine learning approach.

5.1 Classroom Dataset

De-identified data was obtained from a mixed graduate and undergraduate Human Computer Interaction (HCI) course. Most of the course was conducted in a traditional classroom format, however a set of online daily quizzes were incorporated to assess comprehension of the reading material prior to each class. For the purposes of this study, the quiz data are utilized for its temporal and process-level information. As such, the process-level models include these features, while the traditional models created only utilize the summative features. The students that completed the course for a grade included 9 graduate students and 31 undergraduate students (40 total). A variety of student assessment data were collected in the course and a full list is presented in Table 5.1. All data in Table 5.1 were graded on a scale of 0-100, and the remainder of the grade for graduate students was based on an additional term project. The final grade was available both as a raw numeric score on a scale of 0-100, as well as on an ordinal scale (A,B,C, etc.).

Additional grade contribution was available from several additional sources beyond those shown in Table 5.1. These included the graduate project, a case study presentation, course evaluation, and extra credit opportunities. As these were either only available for a small subset of the students or only peripherally related to the subject material, these were excluded from the machine learning analysis. By summing over the column of instances in Table 5.1, the collected dataset represents 27 individual metrics for use in machine learning. For the machine learning prediction algorithms, the final grade (either numerical or ordinal) represents the primary prediction target for supervised learning approaches. In this dataset, the ordinal grade only contained examples of A, B, and C (no D or F), and therefore represent three classes for classification approaches. The remaining 26 metrics were available to be used directly as features in the machine learning models. Additionally, for each category an agglomerative feature was created by taking the average of that category (e.g. "project average"), resulting in five additional features.

Records were also available for excused and unexcused absences for each student, which resulted in a recorded score of "0" for the daily quiz on that day. However, since excused absences were not included in the calculation of the final grade, the five students

Table 5.1: Assessment Techniques used in human factors course dataset. * indicates grade contribution for graduate students

Metric	Type	Description	Number of Instances in Course	Total Contribution to Final Course Grade
Daily Quizzes	Process-Level	Quiz questions regarding comprehension of the reading homework assignments, multiple choice (10 questions)	19	10%,7%*
Projects	Summative	Projects that focused on the understanding and application of the course concepts	3	33%,27%*
Problem Sets (Psets)	Summative	Quantitative homework problem sets	2	12%,8%*
Tests	Summative	Cumulative examinations covering all prior course material	2	40%,25%*

that had excused absences (all students had 2 or more absences) were excluded from the analysis for simplicity in the application of machine learning models. A boxplot of the dataset after removing the excused absentees is shown in Figure 5-1.

The resultant total possible features for machine learning (31, from 26 individual metrics plus 5 agglomerative metrics) is nearly equivalent to the number of individual students (35). This scenario of high number of features to data points poses significant challenges to the machine learning approach [151]. For small datasets, the primary concern is the tendency for machine learning models to overfit training data. Consider a case with 100 unique features and 100 students in the training set. In a linear regression model, a single feature for each student could be used to perfectly predict the final grade. However, the weights associated with these fits would be specifically tailored for the training data and would have difficulty in predicting previously unseen student data.

To counter this tendency to overfit data, feature selection (dimensionality reduction) methods can be utilized to reduce the number of features. In cases where domain knowledge is available, ad-hoc feature selection or reduction can be an effective method [152]. In this case, it is apparent by the number of instances that the largest contributor to feature vector size are the daily quizzes (19 instances). Therefore, one way to approach

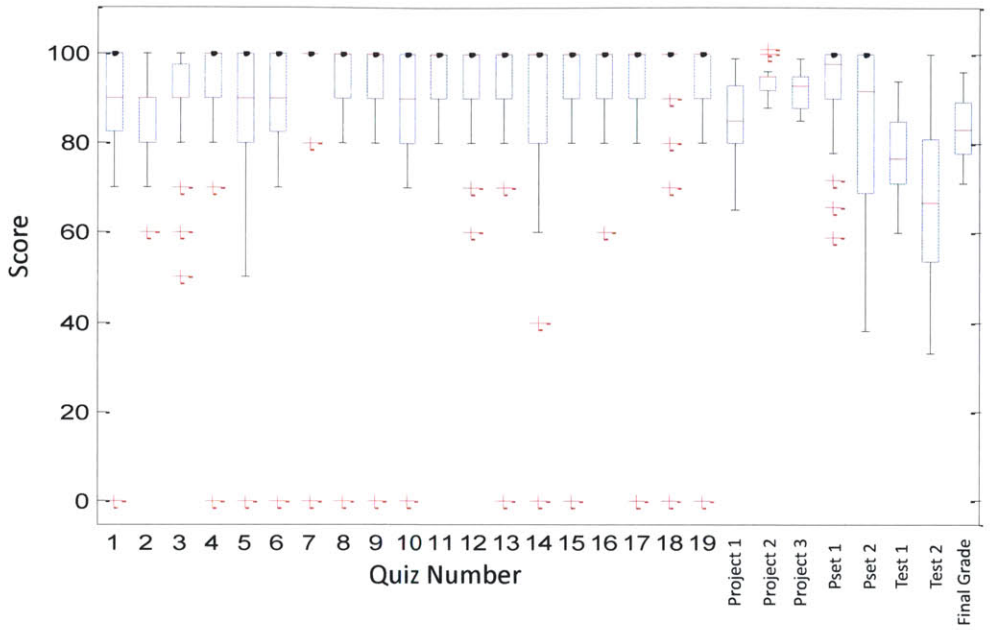


Figure 5-1: Boxplot of dataset after removing excused absentees. Median values are shown as central marks, edges of the box represent 25th-75th percentile. Whiskers are shown for data up to 1.5 times the interquartile range. Outliers are plotted individually with a "+".

the dimensionality reduction is by creating reduced feature sets of quizzes. Unsupervised learning contains a set of algorithms that perform feature selection and reduction by analyzing the relationships between the features and the data. The results of these algorithms on this dataset are shown in the unsupervised learning section below.

5.2 Unsupervised Learning

As discussed in Chapter 2, unsupervised learning techniques can generally be divided into clustering and feature selection methods. Clustering techniques find natural groupings in the data based on similarities across the feature set. Feature selection methods analyze the features themselves to identify useful inputs to machine learning models or create new inputs (through combination or transformation) that describe the variability in the

data with as few features as possible. In this section both of these types of methods are applied to the classroom dataset, and a range of algorithms are tested for their usefulness.

5.2.1 Cluster Analysis

Identifying clusters in data can be useful in helping to identify particular data points that show similar feature values to each other. Thus, cluster analysis can be used to find trainees that exhibit similar behaviors or performance profiles. This can be useful in an assessment and intervention setting, as identifying a set of trainees with similar behaviors could identify a group intervention strategy. However, there are several considerations introduced in Chapter 3 in the application of clustering approaches on a dataset. First, the specific clustering algorithm(s) must be selected. Second, transformations or standardization of the data may be required as cluster algorithms are highly dependent upon measuring distances in the feature space. Third, a particular distance measure for the dataset must be selected. Once these have been addressed, cluster algorithm performance may be compared for the dataset.

Clustering Algorithms

There are a range of clustering algorithms, and a selection of common methods of varying complexity were presented in Chapter 3. As a reminder, these are k-means, hierarchical clustering, Gaussian Mixture Models (GMMs) and Self-Organizing Maps (SOMs). These methods represent a range of clustering algorithm types: a centroid model, connectivity model, distribution model, and topological model, respectively. An analysis of these methods on the classroom dataset described here provides insight into which methods (if any) are useful on data from a typical knowledge-based training domain. Prior to running the algorithms on the dataset, it is important to select a data standardization technique and distance measure as discussed in Chapter 3. The following sections discuss the selection of these parameters for the clustering analysis.

Data Standardization

As discussed in Chapter 3, it can be important for cluster analysis to standardize across different feature types to create consistency in distances measured in the feature space. The two main methods for standardization are the z-score transformation, which fits data

to a standard normal distribution, and the range transformation, which scales values along each feature to a value between 0 and 1. For the classroom dataset presented, both z-score and range transformations were tested for clustering performance using a basic k-means algorithm (excluded here for brevity, see Appendix J for results). For most feature sets, the models created using range-transformed features outperformed those created from z-score transformations. This was particularly observed for feature sets containing the process-level quiz scores, as these values tended to often carry values of 80-100 or 0 when an unexcused absence occurred. With unusual value distributions such as this, it is not surprising that the range transformation produced better performance. Based on these results, in the following sections the analyses are based on models using a range transformation for the input features.

Distance Measure

Chapter 3 also discussed the importance of selecting a distance measure when using clustering algorithms. Since similarity between data points in the feature space is based upon the distance between these points, the choice of distance measure can have a strong impact on clustering algorithm performance. In knowledge-based settings such as the classroom dataset used here, distances are calculated based on the scores from the various assessments (usually graded on a scale of 0-100 as in this dataset). In this sense, the Manhattan distance simply represents the sum of differences in scores across each assessment. As an example, consider a student that has a (pre-transformed) score of 80 on each of the two tests, while a second student received a 90 on both tests. In the Manhattan paradigm, the distance would simply be calculated as $10+10=20$ points difference. The Euclidean distance would treat the two features as orthogonal, calculating a cross-feature distance of $\sqrt{200} \approx 14.14$. Thus, Euclidean distance will calculate a lower distance for differences across multiple features than will the Manhattan distance.

In this simple example, Euclidean distance will treat the difference of 10 across both tests as approximately equivalent to a distance of 14 on a single test. For the classroom dataset, Chebychev distance results in simply finding the greatest difference in score across all assessments (in the example given, this would simply be 10). However, the Chebychev distance would generally not result in a useful measure on this dataset, as all students that had an unexcused absence (and thus a score of 0) would immediately be

assigned a large distance to all students who did not have an absence on that same day. Given that missing a single class is unlikely to be the largest factor in determining overall student performance, it is not prudent to use Chebychev distance on this dataset.

For the classroom dataset, the advantages and disadvantages between using Euclidean distance and Manhattan distance are less clear, but can again be illustrated with an example. Consider a student that gets a score of 80 on all assessments compared to one that gets 90 on all assessments. As the number of features used in the feature space increases, the distance as measured by Manhattan distance will increase linearly. That is, increasing from 10 to 20 dimensions will double the distance. This has the advantage of maintaining a clear meaning to the observer as it directly translates into the sum of differences in scores across the students in question. Regardless of feature space dimensionality, an increase on a single score by one student of 1 point compared to the other student will result in an increase in Manhattan distance by 1. However, this may not always be beneficial at high numbers of features. Consider the data set presented here, with two test scores and 19 quiz scores. On feature sets that include the quizzes, Manhattan distance will be particularly dominated by the quiz scores. This may be contrary to the intention of training evaluators, as other scores (such as the tests) may be deemed to be more important than quizzes for judging similarity. In fact, due to the differing contributions to the final grade across the features, it suggests that not all features should be treated as equally important.

The Euclidean distance has a non-linear relationship with increasing features, such that the increase in distance is depressed relative to the linear relationship seen with Manhattan. Consider our example of the 80s and 90s students. Table 5.2 shows the progression of Manhattan and Euclidean distances for this case for 1, 2, 3, and 4 features. As can be seen in the table, the impact of each added feature on the Euclidean distance measure is depressed relative to the linear progression of the Manhattan distance. Given that the largest feature set in the classroom dataset is the quizzes (which have lower grade contribution than other measures), this depression with increasing number of features may be desirable.

Another strategy to increase the importance of higher-value assessments such as tests is through unequal weighting of features in the calculation of distance in the feature space. Classroom datasets lend themselves particularly well to this approach, as the

Table 5.2: Progression of Manhattan and Euclidean Distance For Example Case

Distance Measure	Number of Features			
	1	2	3	4
Manhattan Distance	10	20	30	40
Euclidean Distance	10	14.14	17.32	20

intermediate assessments during the course are used in the calculation of the final grade. Thus, the relative contribution of each assessment to the final grade can be directly used to weight each feature in the feature space. In this way, trainees that score similarly on highly weighted features such as tests will be measured as closer in the feature space than trainees who score similarly on quizzes. A simple example can illustrate this distinction. Consider three example trainees, Bob, Sam, and Jenny. Bob scores an 80 on a test and a 90 on a quiz, Sam scores a 90 on the test and an 80 on the quiz, and Jenny scores a 90 on both test and quiz. In an unweighted environment, Bob and Sam will have equal similarity with Jenny, even though Sam differed on the highly important test, while Sam differed on the much less important quiz. With weighting, Sam would be considered closer to Jenny than Bob. There may be special circumstances in which it may be desirable to weight all features equally, but in most cases similarity on important summative measures such as tests is likely more informative than similarity on less significant simpler measures. Based on these arguments, a weighted Euclidean distance using the relative contribution of each feature to the final grade was selected for use in the cluster analysis presented below. For example, each test was 20% of the final grade, while all 19 quizzes together comprised 10% of the final grade. Thus in the weighted format, each test was weighted as equivalent to 38 quizzes. With the algorithms, data transformation, and distance measure selected, the algorithms can now be analyzed for their relative performance. The following section reviews the metrics used in the analysis of clustering algorithms.

Metrics for Comparison

To compare the algorithms, a set of measures of performance must be selected. As discussed in Chapter 3, both external and internal metrics can be used. As a reminder, external metrics use knowledge of target classes (from supervised learning) to represent an underlying "truth", while internal metrics focus on the properties of the clusters themselves and do not require any outside information. In this analysis, the external metrics of

entropy and purity as well as the internal metrics of compactness and isolation are used, which are a range of common metrics used for judging cluster algorithm performance. Additionally, subjective judgment from a subject matter expert (SME) was included in the comparison to provide qualitative insight on which clustering algorithms resulted in better performance. The results for the comparisons of the clustering algorithms is presented in the following section.

Cluster Algorithm Comparison

Each cluster algorithm was run with a variety of feature sets, including summative features only, process-level (quiz) features only, and combined features. Summative features included tests, projects, and problem sets. The only process level feature used in this analysis were the quizzes. Additionally, both individual scores (e.g. quiz 1, quiz 2) and averaged scores (e.g. average quiz score) for each feature type were run. For external metrics, the categorical final grade (A,B,C) was used for reference. Since the external metric calculation is simplest where the number of clusters is equal to the number of classes, three clusters were used in this analysis. Additionally, this had the benefit of allowing for significant membership in each cluster (>6-8), which allows for more meaningful interpretation by a training evaluator. For algorithms with random initializations (k-means and SOM), 1000 algorithms runs were computed and the average value is shown. The results of the analysis are shown in Table 5.3. In this table, "Summative Averages" refers to the average of each of the summative metrics (problem sets, projects, tests), "Summative Individual" refers to the individual scores for each summative metric, "Process-level Averages" uses only the quiz average, while "Process-level Individual" uses each quiz score separately. The final two rows, "Total Averages" and "Total Individual" refer to datasets that include both process-level and summative information, either using only average values or individual scores.

Table 5.3: Unsupervised Algorithm Comparison. For clarity, the highest performing algorithms for each metric are shaded

Features	Evaluation	K-means	Agglomerative Clustering	Gaussian Mixture Model	SOM
Summative Averages	Inspection	Slight cluster ambiguity	Tends to combine classes 1 and 2	Slight cluster ambiguity	Slight cluster ambiguity
	Entropy	1.509	1.52	1.502	1.526
	Purity	0.333	0.399	0.333	0.362
	Compactness	2.959	4.043	4.304	2.953
	Isolation	3.1675	2.989	3.303	3.17
Summative Individual	Inspection	Slight cluster ambiguity	Tends to combine classes 1 and 2	Error between classes 1 and 2	Slight error between classes 2 and 3
	Entropy	1.51	1.574	1.517	1.512
	Purity	0.334	0.365	0.336	0.343
	Compactness	12.164	14.112	13.17	11.946
	Isolation	4.554	8.022	4.914	4.451
Process-level Averages	Inspection	Empty clusters	Tends to combine classes 1 and 2	Tends to combine classes 1 and 2	Moderate cluster ambiguity
	Entropy	Empty clusters	1.547	1.526	1.541
	Purity	Empty clusters	0.423	0.362	0.376
	Compactness	Empty clusters	0.339	0.36	0.331
	Isolation	Empty clusters	1.97	1.97	1.491
Process-level Individual	Inspection	Moderate cluster ambiguity	Tends to combine all classes	Data Incompatibility	Tends to combine all classes
	Entropy	1.513	1.403	Data Incompatibility	1.57
	Purity	0.337	0.211	Data Incompatibility	0.359
	Compactness	25.346	27.42	Data Incompatibility	27.31
	Isolation	12.363	25.467	Data Incompatibility	17.038
Total Averages	Inspection	Slight error for class 2	Moderate error for class 2	Slight error for class 2	Slight error at class boundaries
	Entropy	1.51	1.539	1.509	1.526
	Purity	0.334	0.323	0.33	0.362
	Compactness	4.664	4.95	6.073	4.536
	Isolation	3.315	3.526	2.398	3.481
Total Individual	Inspection	Error between classes 2 and 3	Tends to combine all classes	Data Incompatibility	Tends to combine into 2 classes
	Entropy	1.508	1.545	Data Incompatibility	1.45
	Purity	0.331	0.447	Data Incompatibility	0.281
	Compactness	42.303	46.582	Data Incompatibility	44.004
	Isolation	11.418	30.582	Data Incompatibility	19.749

There are several observations that can be made from Table 5.3. First, as evidenced by feature sets that utilize averages have generally better performance than those that utilize the equivalent individual feature sets. Second, several algorithm/feature set combinations resulted in numerical difficulties, and failed to converge on a set of clusters. Specifically, the k-means algorithm resulted in empty clusters when run only on process-level averages, and the GMM algorithm failed on both process-level and combined individual metrics due to data incompatibilities. The difficulty encountered by k-means implies that for just the quiz average (only a single feature), it was not possible to divide into three unique groups. Given that this was only a single feature, it is entirely possible that the data are not well represented by three groups (a weakness of the feature rather than the algorithm). For the GMM difficulties, duplicated information across several features prevented the algorithm from finding a unique solution. While these duplicated values could be manually removed, this work focuses on the use of machine learning algorithms by training evaluators, who may not be aware of or able to make the necessary modifications to the data. Therefore, the data was left unaltered, and this result indicates a weakness of using GMMs on this type of training dataset. Third, k-means and SOM tend to outperform the other two algorithms used, in terms of subjective inspection as well as by entropy and compactness. Agglomerative clustering generally performed the best in purity, but did not perform as well as the other algorithms for most other measures. This is due to the algorithm's tendency to combine classes together into larger clusters, which is not ideal for evaluation as it reduces the separability of trainees for intervention.

Based on these results and the advantages and disadvantages of each algorithm discussed in Chapter 3, the k-means algorithm is identified as the optimal algorithm on this dataset, both for its simplicity and efficiency, as well as the strong performance of the clusters generated relative to the other algorithms. The following section investigates the properties of clusters generated by the k-means algorithm on this dataset.

Cluster Analysis

Using the k-means algorithm selected, clusters were created over a variety of features, primarily those shown in Table 5.1. To determine the optimal number of clusters (one of the requirements for using k-means as discussed earlier), the Bayesian Information Criterion (BIC) was calculated for a range of number of clusters from 1 to 20. The

results are shown in Figure 5-2, noting that lower BIC is preferable. Based on these results, we can see that 3-4 clusters (or even potentially 6 clusters) have the lowest BIC. Given this, and that there are three grade classifications in the dataset (A,B,C), it was selected that three clusters should be used for k-means modeling. Therefore, the results presented in Table 3 for each feature set remain useful for analyzing the performance of the models.

As previously mentioned, the feature sets that utilized averages generally resulted in better performance than those using individual measures. Additionally, the feature sets that included summative assessments (rows 1-2 and 5-6) showed dramatically better performance by inspection than did those with process-level assessment alone. In particular, the models created using summative averages and combined (total) averages demonstrated the best clustering on the set, particularly as compared to the grade classifications. Interestingly, since the performance of k-means was relatively high even for internal metrics, the selection of three groups and the division of students could have

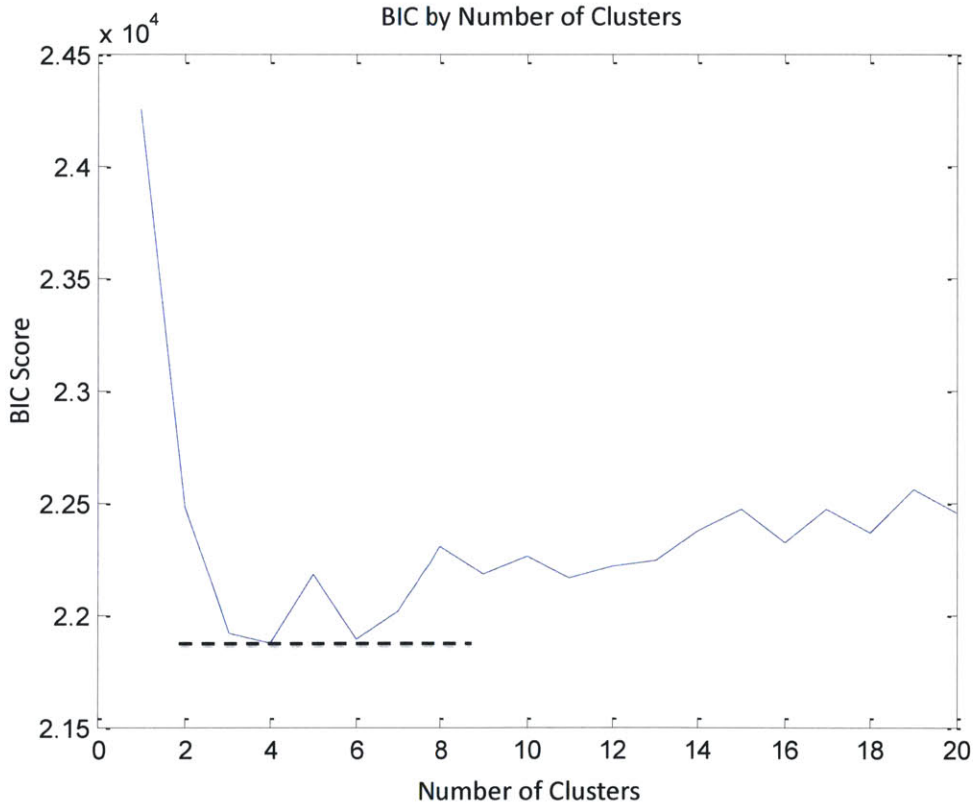


Figure 5-2: BIC analysis of k-means by number of clusters.

been achieved even without the knowledge of the true grade classification. In essence, this methodology lends some objective support to the idea that the selected grade divisions match well with natural divisions in the data. When the clusters are directly compared to the classes, only 2 of 35 students are different between the cluster assignments using k-means and the grade assignments (A,B,C).

Cluster Analysis for Feature Selection

In addition to the feature sets shown in Table 5.3, a cluster analysis was conducted to identify features to be used in supervised analyses presented later in this chapter. Particularly, the number of individual quiz features (19) is the largest contributor to feature set size. If certain quizzes can be selected as particularly informative, a subset of quizzes could be used as features for supervised learning. To this end, the following steps were used:

- Average the quiz score data across all students for each quiz
- For each quiz, subtract this average from 100 to obtain the average error
- Run k-means with 2 groups on the average errors to separate based on higher or lower average error

The results are presented in Table 5.4, for running k-means with 2 clusters. In this way, a set of "high error" quizzes can objectively be identified that exhibit the highest variability. The topics for each of the "high error" quizzes are shown in Table 5.5. Overall, these topic areas may have represented more technical and calculation-intensive areas than the topics quizzes in the low-error group (such as workload, decision making, and situation awareness). By utilizing a subset of the most influential quizzes, we can limit the size of the feature space used in the supervised learning models. Therefore, these features were tested for usefulness in a supervised learning approach. The results of the supervised testing with these "high error" quizzes is presented in the supervised learning section later in this chapter.

Summary

In summary, a set of clustering techniques were tested on the classroom dataset, and k-means was identified as a useful clustering approach. The cluster analysis using this

Table 5.4: K-means clustering results on average quiz error

Cluster	Cluster Center	Member Quizzes
Normal Error	4.05	1,4,7,8,9,11,12,13,15,16,17,18,19
High Error	10.53	2,3,5,6,10,14

Table 5.5: High-Error Quizzes and Associated Topics

Quiz Number	Topics Covered
2	Vision
3	Research Methods
5	Vestibular / Spatial Disorientation
6	Displays
10	Error
14	Controls

technique revealed an independent objective assessment for grade distribution (in this case confirming the instructor's distribution). Additionally, it was shown that clustering methods could be used as part of a dimensionality reduction process that can feed into supervised learning approaches. In the next section, additional unsupervised feature selection analysis is presented, followed by the supervised results.

5.2.2 Dimensionality Reduction

Dimensionality reduction (a form of feature selection) allows for the representation of data in high dimensional space in a lower-dimensional space. There are two primary reasons for this transformation: to remove redundant or irrelevant features, or to try to avoid the curse of dimensionality. The first is fairly self-explanatory, where it allows the user to identify and remove features that are not providing useful content, which helps computation time and can prevent overfitting. The second relates to the idea of the curse of dimensionality, which states that as the number of dimensions increase, the separation between points approaches a constant value. This creates great difficulties for algorithms that require measurements of distances or partitions of space, such as clustering algorithms and Support Vector Machines (SVMs). Typically these issues only arise when the number of features is in the range of 100s to 1000s or more, and thus in this dataset, the first reason (relating to overfitting) is the primary driver for performing dimensionality reduction.

The main strategy for dimension reduction revolves around feature extraction, which

involves the transformation of data from high dimensional space into lower dimensions. The most common technique is Principal Component Analysis (PCA), which performs a linear mapping of the data into lower dimensional space. The objective of PCA is to obtain a smaller set of orthogonal projections along the original feature space in a way that the variance of the data along the new dimensions is maximized. By using the first several components generated, dimensionality reduction can be achieved (for further information on PCA, see [12]). For this dataset, the daily quizzes represent the largest contributor to the dimensionality of the feature space. Therefore, PCA was applied to the 19 quizzes and the detailed results can be found in Appendix K.

The major sources of variation in the dataset are indicated by their high contribution to the first few principle components generated by PCA. While the full principal component vectors are available in Appendix K, Table 5.6 shows the top five quizzes contributing to each of the first three principal components, as determined from their respective coefficient values. It is interesting to note that many of the quizzes that are major contributors to the principal components were previously identified as "high-error" quizzes through the clustering methods. In fact, all of the high-error quizzes appear except for quiz 2. There are also several other quizzes that appear multiple times in the first three components, none of which were labeled as high-error: quizzes 1, 8, and 9. Since these are influential in the PCA model but were not identified as high-error, there must be other aspects to these quizzes that allow differentiation between trainees.

There is an important additional result from the comparison between the "high-error" quizzes as determined by clustering and the results of PCA. In the cluster-based approach, the particular quizzes that are of greatest importance are explicitly identified by the nature of the method. The assessments and topics associated with the high error quizzes suggest to the evaluator topics with which the trainees have difficulty. PCA, on the other hand, requires the evaluator to analyze the resultant components from the analysis to identify the quizzes that make the largest contributions to the component scores.

Table 5.6: Largest Contributing Quizzes to Principal Components. Quizzes that were identified in clustering as "high-error" are bolded.

Component	**Top Five Quizzes**
1	9, **14**, 1, 13, 8
2	**5**, **6**, 17, 4, **3**
3	1, 9, 15, **10**, 8

Additionally, the quizzes (such as identified in Table 5.6 are based on their contributions to the variance in the data, not on the errors committed by the trainees. Thus, while it is seen that there is some overlap between the high error quizzes and the top contributors to the principle components, it would be inadvisable to assume that all of the topics associated with the quizzes in Table 5.6 should be included in an intervention. This means that PCA requires greater interpretation by the training evaluator than the cluster-based approach, and is a disadvantage of using PCA as part of training evaluation.

Overall, this method can be useful to training evaluation in two ways: it can highlight particular assessment metrics (such as the individual quizzes from this dataset) that are particularly descriptive of the data, and the resultant principal components can be used as features in supervised learning prediction of later trainee performance. To determine the effectiveness of PCA as a dimensionality reduction technique on this dataset, the features derived from the first three principal components were used as features in the supervised learning analysis in the next section. The formula shown in Equation 5.1 can be used to calculate the principal component scores for any particular trainee. Note that in this equation, the data X is a 1 x 19 vector, and the weights W is a 19 x 3 vector, resulting in a 1 x 3 vector representing three principal component scores T for each trainee.

$$T = XW \tag{5.1}$$

In the next section on supervised learning, the features generated from PCA are referred to as "PCA-reduced". Since only quizzes were used in the PCA analysis, these features only act as a replacement for the quiz features. Note that in the supervised learning testing, the calculation of features was based on the full PCA result across all quizzes, not just the top five quizzes listed in Table 5.6. In the following section, the performance of the PCA-reduced quiz data is compared to models built using all the quiz information as well as those using just the high-error quizzes.

5.3 Supervised Learning

As discussed in Chapter 2, supervised learning utilizes labeled data to form relationships between features and labels in the data. If the labels represent a state at a future time

compared to the features, this is easily interpreted as a predictive approach. In training, supervised learning can use in-training features to predict later-training or post-training performance. This is useful in several applications as discussed in Chapter 3, particularly for informing trainee intervention (TI) by providing early feedback on trainees that are predicted to have poor performance. This section provides an analysis of how supervised learning approaches can meet the proposed applications in a knowledge-based training setting. The section begins with an overview of the supervised learning algorithms used in the analysis, with a particular focus on the knowledge-based training dataset.

5.3.1 Supervised Algorithms

A range of supervised learning algorithms were introduced in Chapter 2 that cover the most common types of algorithms. As a reminder, these are linear regression, logistic regression, support vector machines (SVMs), and artificial neural networks (ANNs), listed roughly in order of increasing complexity. Linear regression and ANNs are common methods for regression, where the labels are measured on a continuous scale. ANNs are also well suited for classification approaches, as are logistic regression and SVMs. In a classification problem, the labels are discrete. Each of these algorithms were investigated for use on the knowledge-based dataset, in both regression and classification approaches as appropriate. Both regression and classification predictions are useful for training evaluation in knowledge-based settings, as final grades are typically represented both by numeric scores and by letter grades.

For the regression techniques discussed, the numeric final course grade was used as the target for prediction. For the classification techniques listed, the letter grade (A,B,C) instead used as the prediction target. An accurate prediction of continuous measures (such as the numeric course grade) has a higher level of precision over discrete measures such as the letter grade. However, it may not be possible to achieve accurate prediction of the continuous numeric grade using regression techniques. Additionally, it could be argued that from an intervention perspective, the educator does not care about the exact numeric grade and only cares whether the student falls into a category of "needs intervention" or "does not need intervention". Thus for the purposes of informing intervention, the classification approach may be preferable. Due to these considerations, both regression and classification techniques were included in the analysis. The particular methods chosen

represent some of the most common and flexible methods used in machine learning, and have been used successfully in a wide variety of domains [12].

Generalization performance was assessed for all methods using the Leave-One-Out (LOO) strategy described in Chapter 4. For non-ANN methods, the remaining data was used for training the model. For ANN methods (which require verification data for setting model weights), the remaining data was randomly split using 80% training and 20% validation. The use of the LOO strategy helps to mitigate the issues associated with having a small dataset, by attempting to maximize the training set size used in model construction.

Both SVMs and ANNs have parameters that must be selected in the creation of these models. In SVM models, the choice of kernel (such as linear, polynomial, or Gaussian) can impact the capability to fit functions. In this analysis, the standard linear kernel was implemented to maximize the simplicity of the resultant models. For determining the structure of the ANN models, several heuristics can be used. For nearly all mappings of the input space into the output space, one hidden layer is sufficient to characterize the mapping [148] and therefore was selected to be used for the ANNs created in this study. The optimal number of neurons in the hidden layer depends on several variables including the size of the training set, the complexity of the underlying function, the noise, and the activation function selected [149]. However, general heuristics have suggested that the number of training data should be between 2 and 10 times the sum of the size of the input, hidden, and output layers (e.g. [150]). In this study, given the number of data available and the general range of the input layer, a moderate network size of 10 hidden neurons was selected. While fewer neurons could have been used to meet the heuristic value for larger feature sets, using too few neurons also runs the risk of creating a bottleneck of information in the model, limiting the flexibility of the ANN approach.

5.3.2 Regression Results

As previously discussed, the target variable used in regression approaches was the numeric course grade for the student. This section presents both the results of a post-hoc analysis to analyze the prediction accuracy of each method, as well as a temporal analysis of the benefits of process-level information in the regression setting.

Post-hoc Prediction Results

The selected regression methods (linear regression and ANN regression) were compared based on the model fit of the data on the test set, as measured by MSE (lower MSE indicates better model generalization performance). Table 5.7 shows the average performance results on the test set over 10 runs using a variety of combinations of features with each technique, including both agglomerative and individual metrics. Cumulative grade percentage of features used in model construction are presented in the third column of the table.

The data in Table 5.7 reveals several relationships about the feature sets and the machine learning techniques used. By comparing MSE values within a column, the relative contribution of different feature sets to final grade prediction can be observed. The test metrics provide the greatest predictive fits for the final grade relative to the other individual metrics. It also indicates that for all feature sets except the quiz scores, the linear regression models outperform the equivalent ANN model. For the model using only quiz scores, linear regression showed the worst performance among all feature sets, possibly due to overfitting when using the full 19 quiz score features. However, it is interesting to note that the performance when including the other scores (tests, projects, and problem sets), the MSE improves considerably over the quiz scores alone (22.8 and 288, respectively). This indicates that overfitting cannot fully explain the poor performance of linear regression on the quiz score feature set, as including additional features typically will make overfitting issues worse. A more complete explanation can be found in the combination of the contribution of the features to prediction accuracy and their contribution to overfitting. The equivalent model including all scores except the quiz scores achieved an average MSE of 1.53, which is considerably better than the 22.8 seen when quiz scores are included. Therefore, the full model with all scores (22.8) is able to take advantage of the explanatory power contained in the tests, projects, and problem sets to improve performance over the quiz score only model (288), but shows worse performance due to overfitting from the model that does not include quiz scores at all (1.53).

This result is also seen across reduced versions of the quiz score data using linear regression, with the models that include the other metrics (MSE 1.67 and 1.44 for "high-error" and PCA-reduced, respectively) performing better than the models using quiz information alone (55.6 and 45.1, respectively). Overall, this result points to the im-

Table 5.7: Regression results. * indicates only "high error" quizzes used. "+" indicates multiple features used in model creation. PCA-reduced refers to the first three principal components from the quiz scores as determined by PCA. MSE shown is mean +/- standard error from LOO generalization performance. Shading indicates methods with the best performance across algorithms for each feature set.

Features Used	Number of Features	Grade Contribution	Linear Regression MSE	ANN Regression MSE
Quiz Average	1	10%	40.4 ± 7.02	100 ± 21.4
Test Average	1	40%	7.14 ± 1.91	64.8 ± 38.9
Project Average	1	33%	46.4 ± 8.01	126 ± 66.7
Problem Set Average	1	12%	33.9 ± 6.20	63.9 ± 21.8
Quiz Average + Test Average + Project Average + Problem Set Average	4	95%	2.81 ± 1.06	160 ± 29.5
Test Scores + Project Scores + Problem Set Scores	7	85%	1.53 ± 0.821	158 ± 30.3
Quiz Scores	19	10%	288 ± 104	75.3 ± 19.6
Quiz Scores + Test Scores + Project Scores + Problem Set Scores	26	95%	22.8 ± 14.3	183 ± 45.3
Quiz Scores (*)	6	3.2%	55.6 ± 9.89	104 ± 34.0
Quiz Scores (*) + Test Scores + Project Scores + Problem Set Scores	13	88.2%	1.67 ± 0.676	171 ± 34.0
PCA-Reduced	3	10%	45.1 ± 8.85	401 ± 127
PCA-Reduced + Test Scores + Project Scores + Problem Set Scores	10	95%	1.44 ± 0.740	139 ± 24.8

portance of the test scores and other traditional assessment metrics over the online quiz information in post-hoc prediction. While the highest performing model in Table 5.7 was linear regression using the PCA-reduced quiz set in addition to the traditional metrics, the performance was not much better than the equivalent model without any quiz information (1.44 and 1.53, respectively). Therefore, it appears that either in reduced or un-reduced form, the quiz information does not provide significant benefits to post-hoc prediction. While ANNs are able to make better use of the quiz score information alone, since the process-level quiz information provides little explanatory power over the summative metrics there is little evidence to support the use of the more complex ANN techniques over simple linear regression on this dataset.

An additional important insight that can be drawn from Table 5.7 is the relative grade contribution of the feature set in comparison to the model fit using those features (see Table 1). Since the target variable is a function of the input variables, certain model predictions may be more useful than others by providing additional information about the final grade over the inherent grade contribution. Consider the linear regression models using individual metrics of quiz average compared to project average. Since the project average accounts for 33% of the grade, it would be expected to provide more explanatory power of final performance than the quizzes that only contributed 10% of the final grade. However, the generalization performance of the models based on project average (MSE 46.4 and 126 for linear regression and ANN, respectively) are similar to those based on quiz average (40.4 and 100 for linear regression and ANN, respectively). This indicates that the quiz information provides more explanatory power compared to the contribution of the quizzes to the final grade. Since a principal objective of prediction of trainee performances focuses on the early prediction of trainee performance to inform intervention, the ability of metrics such as quizzes to make accurate predictions when not much of the grade is yet accounted for indicates that these metrics may be useful for temporal prediction, and this analysis is presented later in this chapter.

The subset of "high error" quizzes seem to show the highest proportion of explanatory power to grade contribution among all feature sets. This is an important finding for several reasons. First, this indicates that these "high-error" quizzes are able to capture most of the information contained in the quiz scores while reducing the number of features from 19 to 6, which reduces the potential for the model to overfit the data. Second,

since the "high-error" quizzes comprise a smaller percentage of the final grade, this indicates that the relative explanatory power of these features (MSE 55.6 and 104 for 6% contribution) is high compared to the superset of all quizzes (MSE 288 and 75.3 for 10% contribution). However, the low explanatory power of models based on quiz scores alone indicates that while the quizzes may provide a high explanatory power relative to their grade contribution, a successful model will include other features (e.g. tests, problem sets) in addition to the quiz scores.

The information in Table 5.7 can also be used to compare the use of dimensionality reduction through the determination of "high error" quizzes by clustering and the PCA-reduced quiz scores. As can be seen in rows 9 and 11 of the table, the PCA-reduced quiz information results in slightly better generalization performance (MSE = 1.44) than the equivalent models from the "high error" quizzes (MSE = 1.67). This indicates that there is still valuable information for prediction contained within the lower error quizzes, which the PCA transformation is able to utilize. While this suggests that PCA is a better strategy for prediction accuracy, as discussed earlier it also introduces an additional level of interpretation for training evaluators. From an intervention perspective, it is not only useful to identify which trainees are in need of intervention (prediction accuracy) but also which topics should be included in the intervention. This makes the comparison of PCA and cluster-based approaches to dimension reduction more difficult. PCA seems to provide the best prediction accuracy, while cluster-based reduction allows for easier identification of topics to include in intervention.

When making a recommendation about the best algorithm to use in knowledge-based settings, there are additional considerations beyond just the prediction accuracy. As a reminder from Chapter 3, there are two primary applications for supervised learning in training evaluation: improving prediction models and evaluating assessment metrics. For the first application, the prediction performance is indeed the best indicator for identifying the best algorithm. However, evaluating assessment metrics requires that the evaluator be able to understand the contribution of each feature (metric) to the prediction accuracy. Since the main goal of assessment metrics is to be predictive of operational performance, this information can help the evaluator make decisions on which metrics should be added, modified, or removed for future implementations of the training program. The ability of the supervised learning algorithm to help in this task depends upon the simplicity and

clarity of relationships between the features and targets in the model.

The two algorithms tested, linear regression and ANNs, differ markedly in the clarity of the relationships between features and targets. In linear regression, the weights associated of each feature directly correspond to the contribution of that feature to the prediction value. This makes it clear to the evaluator by looking at the model as to which features had the strongest impact on the prediction. In ANNs, however, information is passed through the hidden layer of the model, which complicates the ability to be able to see direct connections between input and output layers. Features which may be dominant in some neurons may be of lesser importance for other neurons, obscuring the overall contribution of the feature to prediction accuracy. Additionally, since ANNs are typically trained by an iterative method called "back-propagation" (similar to gradient descent), the resultant model and weights will depend upon the initialization of the network parameters. In cases where there is multicollinearity, this can be particularly dangerous for interpretation. Rather than indicating any errors in model construction, an ANN will arbitrarily assign weights to collinear features to meet the appropriate total contribution to the final output. Given two features that provide similar information, the model may assign a much higher weighting to the first feature in one run of the algorithm, and a higher weighting to the second feature in another run. Thus, it would be dangerous for the evaluator to assume from the first run that the first feature is of much greater importance to prediction than the second feature. Based on these reasons, the use of more complex models such as ANNs reduces the benefits of supervised learning for evaluating particular assessment metrics.

Predictions of performance have greater value the earlier they can be obtained. Thus, an investigation of the usefulness of process-level information over time is conducted in the next section. As discussed previously, there appear to be few advantages of ANN models over linear models. There are also disadvantages in utilizing ANNs over linear regression models due to the difficulty in determining a clear link between the created model and the contribution of each input. This is a particularly important consideration for this application, as targeted interventions rely on the educator's understanding of which topics or areas are difficult for students. In ANN models, the relationship between individual features and final grade prediction may be difficult to ascertain, and may limit the usefulness of ANNs in a targeted intervention approach. Therefore, for the temporal

analysis presented in the following section, linear regression models were selected for the further analysis.

Temporal Prediction Results

A second investigation was conducted to analyze the capacity of regression models to make predictions of the final grades as early as possible in the course. At each class session of the course, additional assignments or examinations are completed and thus more information is available to the models. Table 5.8 lists the class sessions for which quizzes, projects, problem sets, and test scores were assigned. For a prediction task at any given timepoint, information would be available from assessments at or below that class number. For example, for predictions at class 12, ten quizzes, one project, and one problem set would be available to use as features for prediction.

When considering the temporal progression of the class, quiz grades are accumulated far earlier in the course progression than the other measures. Therefore, this information may be of assistance in making early predictions relative to those based on the traditional classroom measures. To test the predictive assistance provided by quiz scores, three sets of models were created: 1) a model that only utilized the traditional discrete classroom measures (projects, problem sets, and tests), 2) a model that incorporated all quiz scores as features, and 3) a model that incorporated the "high error" quizzes. Note that the PCA-reduced quiz set was not included in this analysis, as it requires knowledge of all 19 quizzes and thus is of limited usefulness in a temporal setting.

Figure 5-3 shows the comparison in generalization performance (as calculated by MSE) over time between the three models used. In Figure 5-3, the data points indicate MSE over the LOO runs for each model, and the error bars show the standard error for the model performance.

Table 5.8: Timing of availability of assessment information. Numbers indicate the class number (out of 23 classes in the semester) at which an assessment was taken. There were no cases in which multiple assessments of the same type were taken in a single class.

Assessment Type	**Classes with Assessment**
Quizzes	2-7, 9-12, 14-22
Projects	11,14,18
Problem Sets	7, 21
Tests	13, 23

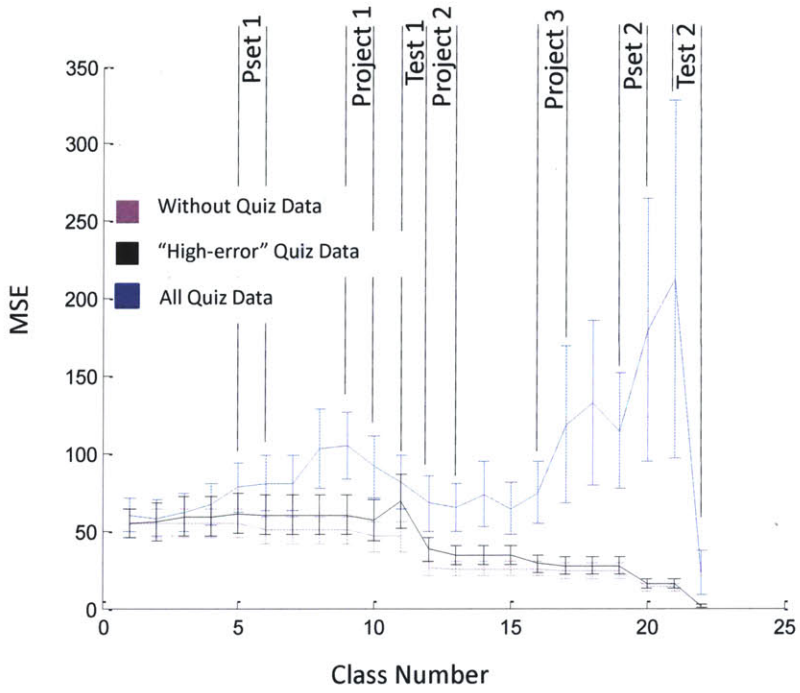


Figure 5-3: Comparison of generalization performance over time of linear prediction models using varying levels of quiz information. Error bars indicate standard error.

There are several important observations from Figure 5-3. First, it can be seen that the trend of the performance across models is the opposite of what would be expected. Specifically, the model with the least available information (without any quiz data), has the best average performance, followed by the model with some quiz information (the "high error" quiz model), and the worst performance is found with the model that includes all quiz data. This result is consistent across all time points in the course, and indicates that more information actually hurts the model performance. However, there is a clear explanation for this: overfitting. While it is counterintuitive that additional information would hurt performance, the model performance shown in Figure 5-3 represents generalization performance on data that was not used to train the model. Thus, if additional information included in a model is used to fit the noise in the training set rather than the true underlying relationships, the generalization performance can decrease. It appears that this is a severe problem when including all quizzes, as these models show

much higher error than either the no-quiz or "high-error" quiz model at the later parts of the course. The use of the "high-error" quizzes helps to mitigate this problem, and has similar performance to the no-quiz model. The overfitting problem is created through the small dataset size associated with this dataset (only 35 trainees), and indicates that in these cases the inclusion of the higher frequency process-level quiz information is not helpful for future prediction, even early in the course when few other metrics are available.

Second, the error bars on the data for the all-quiz models indicate relatively high variability in the models. Since these were generated based on a LOO procedure, this indicates that there is very high sensitivity of these models to the particular datapoint that is being left out of the model training process. This can be further interpreted that there are data points in the data set that are unique; that is, the rest of the data set is not representative of similar behavior to that data point (and thus models are unable to correctly predict the behavior of the unseen data point). This is another limitation of a small data set: if too few data points are included, there is a higher chance that the dataset is not representative of the true distribution of the underlying population. Additionally, the error bars are large enough that it is difficult to unequivocally argue that the inclusion of quiz information hurts performance. For many of the stages of the course, the model performance across the feature sets are similar. Thus, while it is clear that the inclusion of all of the process-level features (the all-quiz model) does not appear to help with early prediction performance, models with some quiz information (such as the "high-error" models) perform similarly to the no-quiz model.

It is clear from Figure 5-3 that overfitting is a problem on this data set even at the early stages of the course, and the modeling approaches are severely limited by the size of the collected dataset. In other CBT settings, such as massive open online courses (MOOCs), this problem may be alleviated through the availability of thousands or more students. This then raises the question of whether process-level information (such as the "high-error" quiz model) is advantageous when the data set size is not so limited, and the data set is truly representative of the underlying population. To address this question, an additional analysis was conducted that investigated the potential improvement using the process-level quiz information through model fit.

Overall Temporal Prediction

While generalization performance is the standard for reporting prediction performance, an alternative strategy can indicate the potential predictive capacity of features through looking at overall model fit (including both training and test data). Since it includes the training data, models created in this way should not be interpreted as indicative of true prediction performance. However, they do indicate the potential information content in the features, assuming that the data set size is essentially unlimited and thus the training and test set are virtually identical. In this way, the potential for these techniques when the dataset size is not as constrained can be investigated.

Figure 5-4 shows the overall model performance for the same three feature sets over time as previously shown in Figure 5-3. In Figure 5-4, the performance of the models is objectively assessed by Sum-Squared Error (SSE) of the model prediction (a variant of MSE), given in equation 5.2. On this dataset, both the MSE and SSE are highly skewed distributions (non-normal) over the LOO model runs. Thus, in Figure 5-4 the points shown represent median SSE and the error bars show a single standard deviation of SSE.

$$SSE = \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 \tag{5.2}$$

where $\hat{y}$ is the predicted final grade by the model, and y is the actual final grade. Lower SSE indicates closer predictions to the actual values, and thus better model performance. The timing of the problem sets, projects, and tests are labeled on Figure 5-4, and it can be seen that for the traditional model (without quiz data), model performance stays constant until new information from one of these sources is available.

Several observations can be made based on these results. First, the process-level models (that include quiz data) exhibit a lower SSE than the traditional model, which is expected since the feature sets of the process-level models are a superset of that of the traditional model. Second, it is apparent that the improvement in performance varies over time (as measured by the vertical distance between the traditional and process-level SSE), which is indicative that not all quizzes contribute equally to prediction performance. Consider the change in model performances for the all-quiz model between classes 2 and 3. At class 2, the performance is nearly identical to the other two models, and thus the quiz data available at that time (quiz 1) does not make a major contribution to

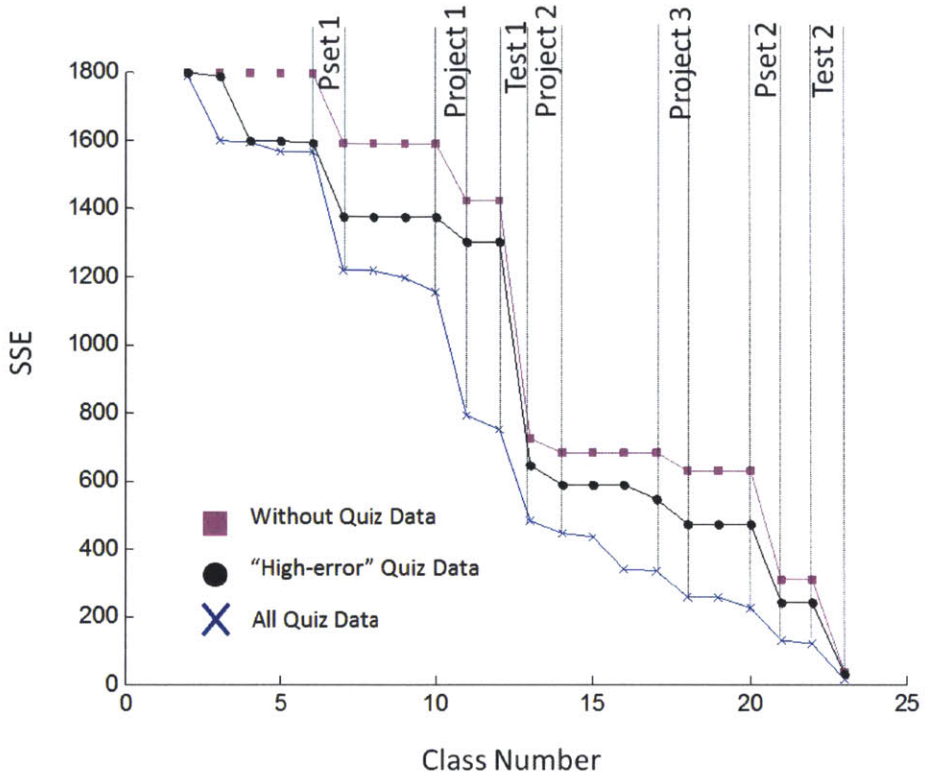


Figure 5-4: Comparison of overall performance over time of linear prediction models using varying levels of quiz information. Error based on predicted final grade compared to actual final grade.

reducing prediction error. By contrast, at class 3 this model improves considerably in performance, indicating that the new set of quiz information (quizzes 1 and 2) allows the model to achieve a much more accurate prediction of final grade. Interestingly, this effect is not solely the result of quiz 2, as the model of high-error quizzes does not see similar improvement at the availability of quiz 2 data. This indicates that it is not merely quiz 2, but the combination of the information from quizzes 1 and 2 that allow the all-quiz model to outperform the others.

Additionally, a comparison can be drawn between the two models that include quiz data in Figure 5-4. As expected, the model with all quizzes is able to achieve a better fit of the data than the model that includes only the "high error" quizzes. For some time periods of the class (e.g., classes 4-6) the two models have very similar prediction performance. However, there are other time periods (e.g., class 11) where the model that

includes all quizzes has a much lower SSE than the "high-error" model.

Based on the results in Figure 5-4, we can also begin to answer the question of when accurate predictions of grade can be made with each of these models. The figure demonstrates that the quiz information improves the prediction over the traditional model, but does this allow for accurate predictions to be made earlier? The determination of how much accuracy is necessary to make a prediction is a complex topic. As has been stated earlier, prediction information has greater value the earlier it is available, but predictions become more accurate with time. Therefore the concepts of model accuracy and time are highly intertwined. Early, accurate predictions have great value for informing targeted intervention approaches. However, the nature of the intervention, cost, and efficacy over time will also play a role in choosing the appropriate model accuracy.

Therefore, the exact performance level necessary to inform intervention is domain dependent. However, we can still draw useful conclusions about the results of the specific temporal analysis in Figure 5-4. A horizontal line on the graph in Figure 5-4 indicates the time at which each model reaches any particular performance threshold. Consider the performance level represented by an SSE of 800. The all-quiz model achieves this level of performance by class 11, while the other two models do not reach this performance level until class 13, just after the first test. There are important implications of this timing difference. The prediction accuracy level that can be achieved after the first test with the no-quiz and high-error-quiz models can be achieved prior to the first test with the all-quiz model. This is very valuable from a targeted intervention approach, as an intervention prior to the first test could improve test performance for struggling students (such as through an extra study session, etc.). In the consideration of both the accuracy and temporal advantages provided by the process-level information, it is clear that this information provides considerable benefits to educators and students.

Overall, the temporal analysis has indicated several important findings. First, the size of the presented knowledge-based data set created major challenges for generalization performance through overfitting. On the presented dataset, the analysis indicated that the few summative level features were more useful for prediction than the process-level quiz data. Second, when the limitations of the data set size are removed, it is clear the process level information does have key performance improvements over the summative-only models, and on larger data sets could be used to inform the timing of training

intervention.

On the classroom dataset, the final grade target is commonly represented as both a continuous numeric value and as a letter grade. Thus, predictions of letter grade could also be useful for training supervisors. On discrete targets such as letter grade, a classification prediction approach is necessary. A similar analysis was conducted for the classification approach and is presented in the following section.

5.3.3 Classification Results

As previously discussed, prediction results of ordinal grade (A,B,C) may be as informative as predictions of continuous grade for the purposes of targeted intervention. If higher accuracy can be achieved through a discrete classification approach on course datasets, this method may be a useful alternative to regression.

Post-hoc Prediction Results

A similar approach was taken to classification as the regression analysis presented earlier. An overall analysis was conducted to determine the best techniques for classification using a variety of feature sets. Table 5.9 shows the results of using a variety of combinations of features with each classification method used, using 10 random data splits (60% training, 20% validation, 20% test) to reduce the importance of random assignment to training, validation, and test sets. In Table 5.9, the standard classification algorithm performance measure of classification error is used. Note, however, that since this is a measure of error, lower values represent better performance. Alternatively a "naive" misclassification rate can be calculated based on simply classifying all students under the class with the highest membership. Since there are 16 students in the largest class in this dataset out of 35 total students (16 "B" grades), this "naive" strategy would result in the misclassification of 19 of 35 students, or a misclassification error rate of around 54%. For some of the feature sets (particularly the quiz-only and project-only feature sets), the algorithms do not perform appreciably better than this "naive" strategy, which indicates that these feature sets have insufficient information to allow for useful application of machine learning classification techniques.

Several conclusions are apparent from the results in Table 5.9. First, when comparing the quiz scores (row 7) to the dimensionality reduced quiz datasets (row 9 for cluster-

Table 5.9: Classification results. * indicates only "high error" quizzes used. "+" indicates multiple features used in model creation. PCA-reduced refers to the first three principal components from the quiz scores as determined by PCA. Shading indicates algorithm(s) with the lowest error for each feature set.

Features Used	Number of Features	Grade Contribution	Logistic Regression Classification Error (%)	SVM Classification Error (%)	ANN Classification Error (%)
Quiz Average	1	10%	55.14 ± 0.56	50.00 ± 0.65	50.86 ± 1.44
Test Average	1	40%	19.43 ± 0.20	23.71 ± 0.23	23.14 ± 1.68
Project Average	1	33%	53.43 ± 0.91	55.43 ± 0.57	50.29 ± 0.76
Problem Set Average	1	12%	44.57 ± 1.05	50.00 ± 0.25	39.14 ± 2.25
Quiz Average + Test Average + Project Average + Problem Set Average	4	95%	6.29 ± 0.78	18.00 ± 1.05	12.00 ± 1.65
Test Scores + Project Scores + Problem Set Scores	7	85%	0.086 ± 0.006	0.114 ± 0.007	0.154 ± 0.031
Quiz Scores	19	10%	27.71 ± 0.65	34.00 ± 0.66	53.43 ± 0.97
Quiz Scores (*)	6	3.2%	43.71 ± 0.85	52.00 ± 0.68	50.29 ± 1.63
Quiz Scores (*) + Test Scores + Project Scores + Problem Set Scores	13	88.2%	13.14 ± 0.69	10.00 ± 0.80	11.14 ± 2.00
PCA-Reduced	3	10%	51.70 ± 0.50	50.00 ± 0.50	51.7 ± 1.20
PCA-Reduced + Test Scores + Project Scores + Problem Set Scores	10	95%	14.30 ± 0.40	14.00 ± 0.40	10.9 ± 1.50

based reduction and row 11 for PCA-reduced), it appears that the full quiz score models generally result in considerably better performance than the reduced models as measured by classification error. Additionally, the PCA-reduced set does not perform better than the "high-error" quiz models from clustering. Both of these findings are in contrast to the regression results, which found that the PCA-reduced feature set provided the best post-hoc performance. Second, no single classification algorithm appears to be consistently superior in prediction performance to the other two. Both logistic regression and ANNs tended to outperform SVM models across most feature sets. This may be due to the use of the standard linear kernel in the SVM algorithm, which will have difficulties modeling non-linearities in the data. For any particular feature set, the resultant ANN model will have a higher complexity than the corresponding logistic regression model. As with regression analysis, the increased complexity and difficulty in drawing connections between inputs and outputs makes ANNs less desirable for informing targeted intervention. Based on the similar classification performance between ANNs and logistic regression models and the greater interpretability of logistic regression, logistic regression was selected for further temporal analysis.

Temporal Prediction Results

For the classification approach, the same set of features over time was used from Table 5.8. At each point in time, three models were constructed: a model without process-level quiz data, a model including all quiz data, and a model including only the "high-error" quizzes. Figure 5-5 below compares the logistic regression model performances over time through classification error.

The data in Figure 5-5 show a similar result to that of Figure 5-3 from the regression results. In generalization performance on this dataset, the quiz information seems to not be helpful in predictions early in the course. The generally worse performance of the models that include the process-level quiz data are likely attributed to overfitting due to the small data set size. Additionally, until after the first test the error rates of all of the models (including the best-performing no-quiz model) are not markedly different than the naive error rate (54%). This is largely due to the difficulty in the convergence of the maximum likelihood estimation calculations, and all of the models in this time range exceeded the iteration limit specified (10,000 iterations). This is an important result as

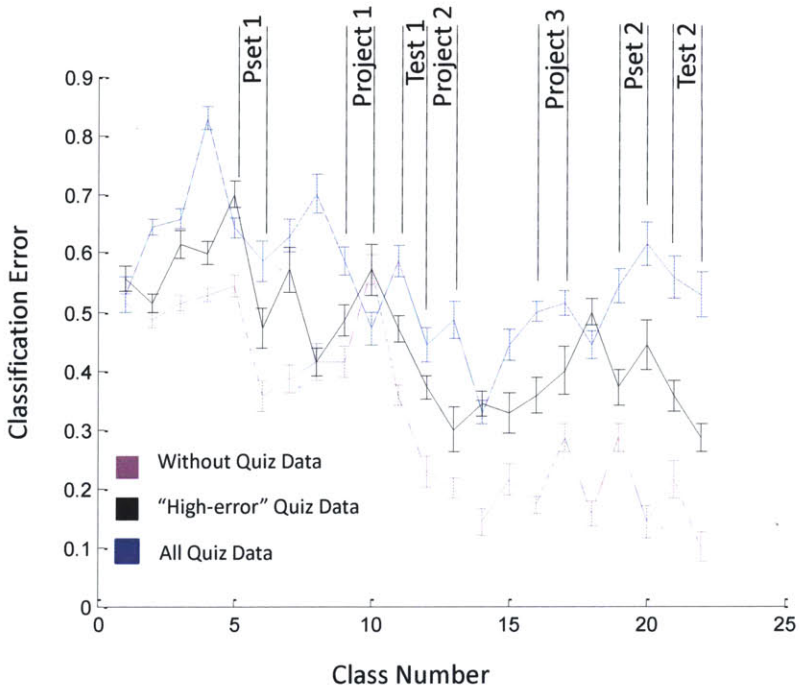


Figure 5-5: Comparison of generalization performance of logistic prediction models using varying levels of quiz information.

well, as the indication is that for few features without high correlation to the targets, the classification approach may not be reasonable.

All in all, it seems that the same overfitting issues that were seen in regression also arise in the classification approach. Similarly to the regression models, it may be worthwhile to view the overall model fit over time to analyze the potential contribution of process-level features in larger datasets (such as MOOCs). The next section presents the results for overall model fit.

Overall Temporal Prediction

A similar analysis was conducted as the regression analysis above, which studied the overall model fit on both training and test data. The results for this analysis are shown in Figure 5-6. In Figure 5-6, we can see that prior to the introduction of the first problem set, the behavior of the models is erratic and is near the naive error rate (54%) for all

three models. This is largely due to the difficulty in the convergence of the maximum likelihood estimation calculations, and all of the models in this time range exceeded the iteration limit specified (10,000 iterations). This is an important result as well, as the indication is that for few features without high correlation to the targets, the classification approach may not be reasonable.

Beyond the first problem set, we can see the models begin to diverge in performance, with the models that include quiz data showing lower classification error than the traditional model. The all-quiz model maintains the best performance, while the "high-error" quiz model has performance in between the all-quiz model and the traditional model. This result is expected as the information included in the model increases from the no-quiz model to the high-error model to the all-quiz model. The models including quiz data show similar performance near the beginning of the course (e.g. classes 6-10), but diverge in performance just before the first test (classes 11-12). At the end of the course,

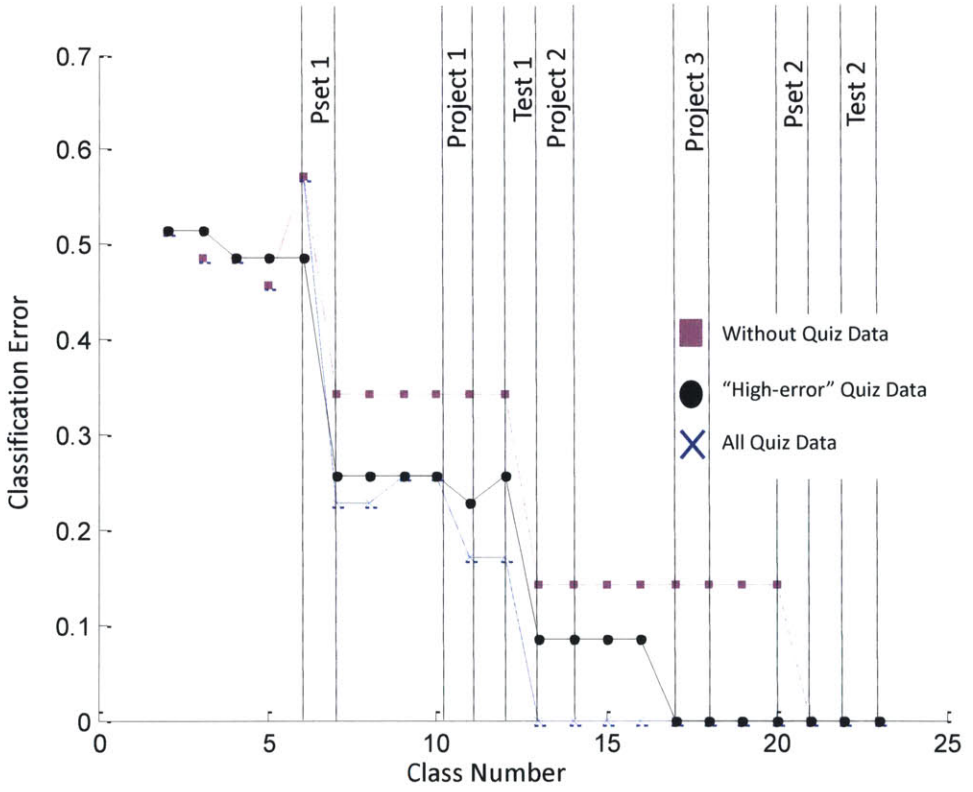


Figure 5-6: Comparison of performance of logistic prediction models using varying levels of quiz information.

the "high-error" quiz model again demonstrates moderate performance, achieving perfect classification before the traditional model but after the all-quiz model.

The results show similar implications to those from the overall model fit regression analysis, such that the models that include process-level quiz information offer potential advantages for both accuracy and timing. For example, the all-quiz model is able to achieve approximately the same classification performance before the first test as the traditional model achieves after the first test, which could be immensely useful for interventions. Additionally, we can note that perfect classification is achievable immediately after the first test with the process level model, while only occurs near the end of the course for the traditional model, indicating that accurate classifications can be made with the all-quiz model 8 classes before the traditional model. The additional time lends these predictions considerably more value over those of the traditional model, as it allows for earlier intervention in the course. From these temporal results, it is clear that the quiz information allows for accurate predictions to be made earlier in the course, providing additional benefits to educators and students. The decision of when an intervention should be applied given this dataset is complex and requires knowledge of the nature of the intervention, and is discussed further in the next section along with other implications of the supervised learning results.

5.3.4 Discussion of Prediction Results

The results indicate several important findings. For post-hoc prediction, the quiz data does not significantly contribute to prediction performance. This is visible from Tables 5.7 and 5.9, where the models constructed using either only the agglomerative quiz average metric or the individual quiz scores did not achieve strong fits of the data. In regression, these models exhibited $r^2 < 0.3$, while for classification these models did not perform better than simple naive models. Other features appear to be far more useful in post-hoc prediction accuracy, such as the tests in this dataset. This provides an important lesson for training evaluators: when summative metrics such as test data are available, these may be more useful to prediction accuracy than process-level metrics.

However, one of the advantages of the quiz data is the increased frequency of these assessments over the summative metrics. On the small dataset presented here, the generalization performances shown in Figures 5-3 and 5-5 indicate that despite the higher

frequency of collection, the process-level features provide no additional benefits in generalization performance. However, this result is likely due to the small size of the dataset which results in overfitting when using the process-level quiz features. While the "all-quiz" model showed markedly worse performance than the "no-quiz" model, it is interesting to note that the "high-error" quiz models had similar performance to the "no-quiz" model, and thus may be a better way to incorporate process-level information without the same overfitting issues. When the overfitting problem is removed (such as in a much larger dataset), it is clear from Figures 5-4 and 5-6 that the models with quiz information (both "high-error" and "all-quiz") provide benefits to improving the accuracy of the predictions made earlier in the course (in both SSE and classification error). In short, while the summative assessment techniques have better overall correlation to the final course grade, the process-level quiz data can improve prediction performance when these other metrics are not yet available. The temporal analysis indicates that process-level information may be very useful to educators to inform targeted intervention, as the earlier an intervention can be made the greater the potential impact on the outcome of the course.

The temporal analysis for overall model fit also indicates that the advantages are greatest when using the full set of process-level information (all quizzes). The use of "high-error" quizzes (to reduce the feature space) appears to improve performance over the traditional model but does not provide as strong benefits as the inclusion of all quiz information through time. This is particularly prominent in Figure 5-4, where the performance of the all-quiz model is much better than the "high-error" quiz model just before the first test. This result has important implications for intervention on this dataset, as it means that students in need of intervention could be identified prior to the first major summative assessment. However, due to the overfitting issue seen in the generalization performance, the advantages of the entire process-level feature set may only be realized on much larger datasets.

The analysis also allows for the comparison of the regression and classification approaches. Both Tables 5.7 and 5.9 indicate that summative information (e.g. tests or test average) is important to have strong post-hoc performance, and there may be little difference between the approaches when both summative and process-level data are available. A more interesting comparison is available in the overall temporal analysis, through the comparison of Figures 5-4 and 5-6. In both cases, error is observed to

decrease with time. However, the progression of prediction errors are quite different. Consider the prediction values for the all-quiz model just before and after the first test. In the classification approach, the error drops from 17.14 (misclassifying 6 students) to 0. Qualitatively this seems to be a large change in prediction accuracy. In the regression approach, the SSE drops from 751 to 483. If we extract the average error from this value (by dividing by 35 students and taking the square root), we can see the average error per student change from 4.63 to 3.71. This difference seems far less important, as it seems unlikely that this average error change would drastically affect the educator's targeted intervention planning. This indicates that the classification approach experiences greater changes in prediction accuracy through time and more rapidly approaches near-perfect prediction. Regression, on the other hand, progresses more slowly in prediction accuracy, and this indicates that predictions earlier in the course may hold similar accuracy as those made later in the course. Therefore, it appears that there may be advantages to the classification approach later in the course, while regression may be more consistent in performance earlier in the course. From an intervention perspective, on this dataset the classification approach is preferred to regression when high certainty of trainees needing intervention is needed (such as a high cost of intervention), while regression would be preferred if the evaluator wants to identify trainees in need of intervention earlier in the program. Ultimately, the utility of either approach will likely depend upon the particular training context.

However, there are several important caveats to these results: the determination of what model accuracy is needed and the exact timing and nature of any intervention is dependent upon the domain. Therefore, it is necessary to have subject-matter experts (ideally, the educators themselves) provide interpretation to the prediction results. These experts have knowledge of the potential benefits and costs associated with an intervention, and can apply this knowledge to the rates of model accuracy over time. Consider the data shown in Figure 5-6. If the cost of an intervention is very high relative to the cost of a failed student or trainee, it would be appropriate to wait until the model is able to achieve near-perfect classification prior to performing an intervention. This would result in the earliest possible intervention timing to be class 13 with the all-quiz model, class 17 with the "high-error" quiz model, and class 21 with the traditional model. Under different assumptions of the cost/benefit properties of intervention and the costs associated with

having a student finish the course with poor performance, the appropriate time for intervention requires interpretation. However for this dataset, under all circumstances, the process-level quiz data allowed for greater certainty in intervention earlier in the course.

For both regression and classification analysis on this dataset, simpler models (linear regression and logistic regression) were able to achieve similar prediction performance to more complex models (ANNs and SVMs). This may indicate simpler (or in particular, linear) relationships between the features and the targets for this dataset. For more complex datasets, it is yet to be shown whether models such as ANNs that can handle non-linearities will perform stronger in a similar prediction task. In the selection of particular methods for regression and classification on the presented dataset, it is important to remember that one of the primary goals of modeling work is to preserve parsimony, hearkening back to principles such as "Occam's Razor". With this in mind and the similar performance of linear and logistic regression to more complex models, the results clearly recommend the use of linear regression for regression analysis, and logistic regression for classification.

5.4 Summary

This chapter described the selection of appropriate machine learning algorithms on a knowledge-based training dataset, and the results of the application of these methods for both unsupervised and supervised learning approaches. There are several main takeaways from this work:

1. The performance and simplicity of k-means make it a better fit for the knowledge-based classroom dataset among clustering techniques. Summative measures proved more informative than process-level measures for clustering.

2. Unsupervised learning provides for dimensionality reduction prior to applying supervised learning on the dataset. In this case the reduced set of quizzes based on cluster analysis provided similar post-hoc performance in prediction through supervised algorithms compared to the PCA-based dimensionality reduction. However, the cluster-based "high-error" quizzes provide more direct insight into which topics or modules that are difficult for trainees than PCA-based methods, and thus may be more useful from a quality control perspective.

3. Unsupervised learning can provide objective feedback to instructors on categorical splits of student or trainee performance, by providing an optimal number of performance clusters and the appropriate cluster assignments.

4. The use of process-level information on the small dataset presented did not improve summative prediction either post-hoc or temporally (the "high-error" quizzes maintained similar generalization performance), but an analysis of the potential benefit of the process-level information in a larger dataset indicated that it could provide better accuracy predictions earlier in the course either through the use of all process-level information or the reduced "high-error" feature set.

5. Simpler supervised learning algorithms are able to achieve similar or better generalization performance to more complex models, indicating that methods such as linear and logistic regression may be preferable on these types of datasets.

This chapter has presented findings from machine learning approaches on an important training setting: a knowledge-based training environment. The focus of knowledge-based training on the development of high-level cognitive knowledge and skills, the lower dimensionality (as compared to the rule-based dataset) and the lack of repetition of assessments influenced the utility of the machine learning algorithms. There are both similarities and differences between the findings in this chapter compared to the previous chapter, Chapter 4. The comparison of the rule-based and knowledge-based settings and the overall lessons for training evaluators are discussed in the following chapter, Chapter 6.

Chapter 6

Comparison of Training Environments

Chapter 4 and Chapter 5 described the process and results of applying machine learning algorithms to training datasets from rule-based and knowledge-based environments, respectively. Each investigation provided insights into the difficulties and potential applications of machine learning approaches for each environment. The datasets from the two training domains have distinct properties which impacted the required data preparation and the results of the machine learning analysis. The rule-based dataset was notable for its low number of modules (3) and the relatively high number of process-level features (372). In some respects, it could be considered a more complex dataset due to the difficulty of adherence measurement and the action-level detail of the adherence metrics collected. The knowledge-based dataset represented a longer training program, but the process level information contained in the quiz scores was not as detailed as that of the rule-based dataset. The properties of each dataset present unique challenges to the use of machine learning algorithms, and this chapter discusses the findings from each analysis with respect to the training environment and training evaluation.

This chapter serves to compare and contrast the results from Chapters 4 and 5, discussing applications of machine learning in other datasets from similar environments and to general training evaluation. It is divided into three main sections. The first section provides a comparison of results between the two environments with respect to each application. The second section covers the implications of the findings from this research for training evaluators. The third section presents the limitations of the present

study, and additional considerations for the interpretation of the results.

6.1 Applications of Machine Learning to Training Evaluation

Chapter 3 proposed that there are six primary applications of machine learning algorithms to training evaluation, presented in Figure 3-2 and reproduced here as Figure 6-1. These applications were generally divided based on the use of unsupervised learning, supervised learning, or a combination of both strategies. It was proposed that unsupervised learning could contribute to label definition and the identification of high-error modules/steps, supervised learning could provide improved prediction models and help assess trainee evaluation methods, and the combination could improve feature selection for prediction and inform intervention timing/content. Each of these applications are reviewed below, in the context of the findings from the two environments presented in Chapters 4 and 5.

6.1.1 Label Definition

The first application of unsupervised learning techniques was to assist training evaluators with label definition. As a reminder from Chapter 3, label definitions (such as the determination of a cutoff value to give a "pass" vs "fail" to trainees) can be difficult to identify. Most commonly, these definitions are based upon subjective judgment by training evaluators or chosen to split the trainees into groups of specific sizes (e.g. give the top 20% of the class an "A"). These subjective judgments may not result in optimal cutoff values, such that either good trainees do not pass the program, or poor trainees are able to do so. There could be considerable debate over which of these scenarios is preferable to the other, and it will typically depend upon the domain. If the operational environment is hazardous or in which an operator error could have large economic or safety implications, it is likely that the program would want to favor being conservative in graduating trainees, with only the best trainees allowed to enter the operational environment. Alternatively, when the training program is long and expensive, attrition of trainees can result in a significant loss of resources, and it may make more sense to let marginally performing trainees enter the workforce. Often, the domain may contain elements of both of these

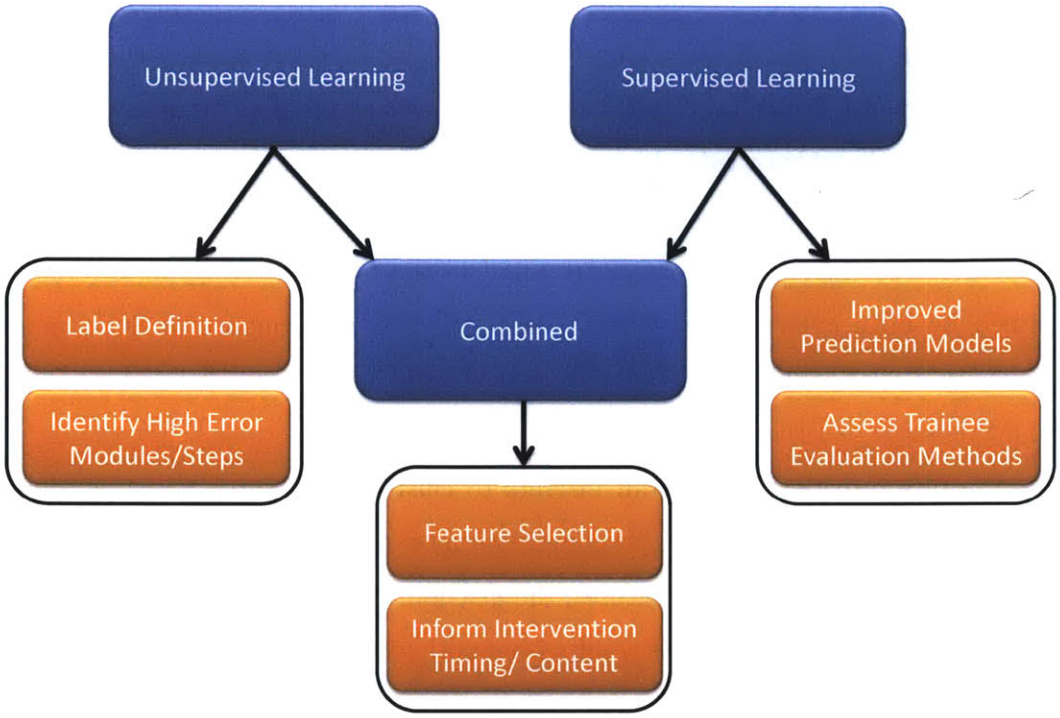


Figure 6-1: Hierarchy of Potential Machine Learning Applications to Human Training by Algorithm Type.

factors, and the benefits of each strategy must be weighted. However, since these will be domain specific, this work cannot make general recommendations across all domains in this area.

Regardless of the optimal strategy, it is advantageous to select cutoff values that provide clear splits in the performance data. When there are few trainees with performance near the cutoff value, there is less chance of trainees being on the incorrect side of the cutoff. This is the advantage of using unsupervised learning for informing label definition. By the nature of the algorithms, they attempt to find clusters that are far apart (high isolation) and where within-cluster distances are small (high compactness). In order to fulfill these requirements, they tend to find natural splits in the data, in which the separation of points in different classes is high. Choosing a cutoff with high class separation may be apparent in simple cases, but in high dimensional space with unusual distributions of trainee performance it may not be as clear to find natural splits in the

data. Clustering methods objectively determine good places for these cutoffs, and thus can inform the selection of cutoff values for labeling by training evaluators.

The results from Chapters 4 and 5 not only provide an evaluation for the effectiveness for the clustering methods on each dataset, but also which clustering algorithms provide the best cluster performance (and thus most confident selection of labels on the data). In both the rule-based and knowledge-based datasets, there was a wide range of performance across the different clustering algorithms. Specifically, the simpler methods (k-means and agglomerative clustering) generally resulted in the best clustering performance as determined by both external and internal metrics. The performance of SOM models were typically similar or slightly worse than these simpler methods, while the distribution-based GMMs struggled the most with both datasets.

These results provide some overall insights to clustering on these types of training datasets. First, the poor results of GMMs suggest that even with the data transformations to try to standardize the different features, the distributions over features are not well fit with normal distributions. Since this results from both environments, it suggests this may be an endemic property of human training data. Assessment data from many training environments may not fit a normal distribution well, either due to the noise associated with human behavior or due to the nature or difficulty of the task. Thus, clustering methods that are agnostic to distribution are likely more flexible to deal with these types of training datasets.

Second, there appear to be simple, high-level groupings that emerge amongst trainees that are able to be captured by the basic k-means and agglomerative clustering algorithms. While SOMs (which make use of ANN architectures) are able to have higher flexibility to fit complex structures within the data, it appears that these structures do not exist within these datasets, or at least are not significant enough to improve the performance of SOMs over the more basic algorithms. It should be noted that this may relate to the small size of the datasets, and it is possible that with larger numbers of trainees more complex structures within the data would begin to form. However, as discussed in Chapter 3, small dataset size is one of the principal constraints of current training evaluation, and understanding the properties of algorithm performance on these typical dataset sizes is important.

However, the results across both environments differ in terms of the usefulness of

unsupervised clustering techniques for the selection of labels. For the knowledge-based training dataset, clustering techniques were able to determine that three groupings were appropriate, and the cluster assignments of trainees was a close match to the final grade assignments ("A","B","C"). This agreement lends support to the idea that the clustering techniques are mimicking the decision making process of the training evaluator, and that the evaluator made judgments for grade assignments that are in line with the objective recommendations provided by the clustering algorithms. However, it is important to remember that while there is good agreement on this dataset, fundamentally the cluster results do not assign order to clusters, while the grade categories are by their nature ordered. Thus, clustering for the purposes of label definition should be done with care in cases where the labels have a natural ordering.

On the rule-based dataset, there were no natural performance groupings and thus no baseline for comparing labels provided by unsupervised learning techniques. The BIC indicated that that only a single performance group existed across all performance metrics, but this result has little practical value for training supervisors for informing trainee intervention (TI). While an alternate method ("elbow method") provided another means for defining labels, these labels cannot be directly translated into performance categories on this dataset. An additional analyis using the 4-dimensional test module performance vector did indicate that these cluster could be loosely interpreted as overall "poor", "moderate" and "good" categories for labeling. Without the knowledge of the actual performances, it would be difficult to assign these qualitative labels to the groups.

6.1.2 Identify High-Error Modules/Steps

The second application of unsupervised learning was to help training evaluators identify high-error modules or steps. As discussed in Chapter 3, the nature of computer-based training data may create cases where a high number of features are available, but the dataset is still heavily limited in number of trainees. In these cases, dimensionality reduction is critical prior to any prediction to prevent overfitting. It operates by identifying current features or transformations of features that have high separation of the data in the feature space. This has the additional benefit for training evaluators that it can identify important modules or assessments that provide the greatest insight into trainee performance (such as high-error modules/steps), and may be the most important topics

to be included in TI.

For the knowledge-based dataset, a k-means clustering approach was used to separate quizzes into "low error" and "high error" based on trainee performance. This was able to identify a set of six "high error" quizzes that had greater overall performance separation between trainees than the "low error" quizzes. This subset of high-error quizzes could be utilized in two ways: as a form of dimensionality reduction (discussed later in the section on feature selection), and to assist with the identification of potential topics for intervention (discussed later in the section on intervention timing/content). Overall, on the knowledge based dataset clustering was able to efficiently identify high error quizzes that could be utilized in training evaluation. In addition to topics of intervention, these high error quizzes could be targets for redesign by the evaluator to improve performance in future iterations of the training program. However, the designer must be careful to avoid improving training performance without improving comprehension and transfer, as discussed in Chapter 2.

In the rule-based dataset, the equivalent use of k-means to separate process-level action data into "low error" and "high error" was difficult due to the high dimensionality of the process-level feature space and the nature of the data. As discussed in Chapter 3, the curse of dimensionality reduces the effectiveness of clustering algorithms in sparse feature spaces, which may have contributed to the difficulty of utilizing this method on the rule-based dataset. Additionally, the initial cluster comparison through the Bayesian Information Criterion (BIC) indicated that the performance data across all variables may be best represented by a single cluster. In this case, it would be difficult to separate the data into "high error" and "low error" steps, and may have contributed to the difficulty of this approach on the rule-based dataset.

Due to the differing success of this approach in these two environments, it can be concluded that the usefulness of this application of unsupervised learning may highly depend upon the dataset (and therefore domain). In particular, it seems to have been useful for the lower-dimensional knowledge-based dataset, while not on the high-dimensional rule-based dataset. This could indicate that this approach is best utilized only when utilizing few features (such as module-level features) rather than individual process-level features. In other words, this strategy may be better at identifying overarching topics with which the trainees have difficulty rather than individual actions.

6.1.3 Improved Prediction Models

The first proposed application for supervised learning algorithms to training evaluation was in the creation of improved prediction models. From a training evaluation perspective, these models have two important uses. The first is that strong prediction models could provide a prediction of future trainee performance, which is a key goal of training evaluation. An important part of developing strong prediction models includes selecting the appropriate modeling techniques and feature sets. Second, predictions of trainee performance later in the course could provide the basis for identifying trainees that are in need of intervention. The first is addressed through the post-hoc analyses conducted in Chapters 4 and 5 and is discussed here, while the second through the temporal analyses in those chapters and is discussed later in the section on informing intervention timing/content.

The post-hoc analyses in both knowledge-based and rule-based environments provide an indication of which supervised learning strategies (regression vs. classification) and which algorithms (e.g. linear regression) provide the highest prediction performance on these datasets. By comparing algorithm performance across a variety of feature sets, the best modeling approaches for trainee performance prediction can be identified. Note that these post-hoc analyses required full datasets over each training program, and thus could not be implemented on an incomplete training dataset.

In regression analyses on both datasets, the simpler linear regression algorithms either performed equivalently or better than the more complex ANN regression models, as measured by MSE. This is a similar finding to the unsupervised results: fitting the basic relationships contained in the data is sufficient for prediction purposes. While there may be more complex underlying relationships in the data, the dataset size is not large enough for the ANN models to have an advantage over the simple linear models. Additionally, due to the structure of ANN models it is much more difficult for an evaluator to determine which features were the major contributors to the prediction value, which is an important aspect of assessing evaluation methods. For the training evaluator to understand which assessments are the best predictors of final performance, the relationship between the features and targets must be clearly defined. Thus, even if with larger datasets the complexities in the data allow ANN methods to provide a slightly better fit than linear regression, it is unlikely that the improvement in model fit would justify the increase in

difficulty of interpretation.

For classification, a slightly different picture appears. In the knowledge-based dataset, the same result arises that favors the simple method (logistic regression for classification) over SVM and ANN methods. However, for the rule-based dataset, numerical difficulties for logistic regression suggest that the flexibility of ANNs is able to better handle the more complex dataset. Despite the preference for ANNs, results indicated that the classification approach in general struggled on the rule-based dataset, only providing marginal gains in performance over "naive" prediction models. It is possible that the difficulty on this dataset arises from the complexities of measuring adherence and other performance metrics in rule-based environments, which suggests that the development of novel metrics could help to improve prediction performance on these datasets.

6.1.4 Assess Trainee Evaluation Methods

The second application of supervised learning algorithms provides a quality assurance aspect to the evaluation process by providing feedback on the assessment metrics collected during the training program. The goal of assessment metrics is to be predictive of future performance, and thus those metrics that are not predictive may be candidates for modification or removal from future iterations of the training program. By comparing the contributions of the relative feature sets in the post-hoc prediction analyses in Chapters 4 and 5, it is possible to identify metrics that provide the greatest prediction performance in each training environment. In particular, this analysis can be used to investigate the contribution of the new process-level metrics enabled in computer-based training settings in comparison to traditional summative assessment metrics.

The importance of process-level information in post-hoc prediction performance differed across the two environments. In the rule-based dataset, summative information provided the largest contribution to prediction performance, and the inclusion of process-level features did not improve prediction performance for both adherence and power output target variables. However, the summative (module-level) form of adherence provided the strongest post-hoc prediction accuracy. Since adherence was required to be measured at an action-by-action level, even though the process-level features were not useful in modeling, the results suggest that the gathering of process-level information is useful in rule-based environments. However, for the knowledge-based dataset, summative features

(particularly the test scores) provided the majority of post-hoc prediction performance. When these features were included in the models, the addition of process-level information did not further improve prediction performance.

These results provide mixed evidence for the importance of collecting process-level information to support training evaluation. The findings suggest that when the summative measures provide strong post-hoc predictions of final performance, the addition of process-level information may not provide significant benefits. However, when the post-hoc prediction performance using summative information alone is relatively weak (as was seen in the rule-based dataset), process-level information can be useful to improve the models when calculated at the module-level. This result could be extrapolated even further, such that in well-established training programs where the assessment methods are well understood to relate to final performance, collecting additional process-level information may not have benefits for post-hoc prediction. In newer training programs where the assessment methods and their relation to final performance is not well understood, it may be useful to build process-level assessment metrics into the training program. Additionally, there may be benefits to the trainee by incorporating high-frequency process-level assessments into the training program. Beyond these post-hoc findings, the temporal analyses conducted also provide insights into the relative importance of process-level information.

6.1.5 Feature Selection

The first combined application of unsupervised and supervised learning utilizes the dimensionality reduction techniques to prepare appropriate features prior to supervised approaches. This is a particularly important application for training programs that utilize CBT technologies to gather larger and more complex datasets. In particular, the availability of process-level information has the potential to improve prediction performance as described earlier, but can create issues of overfitting by increasing the dimensionality of the feature space if used directly. Therefore, the analyses in Chapters 4 and 5 investigated the use of several dimensionality reduction approaches and the impact on prediction performance of supervised learning models created using the reduced feature sets.

On the rule-based and knowledge-based datasets, the primary method of dimensional-

ity reduction was principal components analysis (PCA). Additionally for the knowledge-based dataset, k-means was used to separate quiz scores into "low error" and "high error" quizzes. These reduced datasets were then used as features for the supervised analyses. In the knowledge-based dataset, it was possible to include the full quiz score data as a comparison to the reduced quiz set. However, in the rule-based dataset the process-level data based on individual actions was too high dimensionality for direct use in supervised learning, so no equivalent comparison was possible. Additionally, with only three training modules, the separation of module-level performance into "high-error" and "low error" was not appropriate for this rule-based dataset but could be utilized on longer training programs.

The comparison from the knowledge-based dataset indicated that the models built on the reduced set of quiz information was able to capture much of the post-hoc prediction accuracy achieved by models built from the full quiz information. However, in a temporal analysis, the reduced quiz feature set was not able to achieve the early prediction results obtained by using the full model. This indicates an important lesson for machine learning on these datasets: information is lost during the dimensionality reduction process. While this may be obvious based on the smaller number of features after the process, it is important to recognize that this has downstream impacts on the prediction performance using these features. It can be a useful way to condense information in cases with limited number of trainees, but is not as effective as using the full feature set when possible.

6.1.6 Inform Intervention Timing/Content

The second combined application of unsupervised and supervised learning focuses on TI. In particular, supervised learning algorithms that predict performance later in the training program could be used to identify struggling trainees early in the program, such that interventions can be the most effective. Additionally, the use of unsupervised techniques (through the identification of high error modules discussed earlier) can suggest possible topics for the intervention. The temporal analyses in Chapters 4 and 5 investigate the ability of machine learning approaches to address this application.

The results from the temporal analysis in generalization performance were generally in agreement across the two datasets collected. Due to overfitting, the prediction accuracy on previously unseen datapoints was not improved by the inclusion of process-level infor-

mation. While this strongly indicates that the process-level data should not be included in prediction models using such small datasets, the usefulness in the summative adherence information in the rule-based dataset do suggest the collection of process-level metrics. The lack of early prediction benefits for the rule-based environment could be compounded by the shortened training program, which only included three modules. With a longer training program and additional trainees, it is possible that the process-level information could be useful.

When the limitation of a small dataset is removed the importance of the high-frequency availability of process-level information changes considerably. When looking at overall model fit on the knowledge-based dataset, it is clear that there are significant potential benefits from using the process-level information on prediction performance part-way through the training program, even though there was little performance increase over summative-only models for post-hoc predictions. This suggests that for larger datasets (such as from MOOCs), the availability of process-level information could allow training evaluators to identify trainees in need of intervention much earlier than without this information.

6.1.7 Summary

The previous sections have compared and contrasted the findings from the rule-based and knowledge-based datasets. Table 6.1 below presents a summary of these findings. Overall, the use of machine learning algorithms had greater success on the knowledge-based dataset. This may be due to the shortness or the relative complexity of the rule-based dataset as compared to the knowledge-based dataset. However, it was demonstrated that machine learning approaches could be used to improve training evaluation across both training domains. There are several lessons that have been learned in the process of the analyses presented in this work that could assist training evaluators in future uses of machine learning approaches. The next section presents these implications of the findings for training evaluators.

Table 6.1: Summary of Comparison between Rule-based and Knowledge-based Datasets

Application	Rule-Based	Knowledge-Based
Label Definition	Difficulty defining labels for continuous metrics, related to low separation of performance data	Able to objectively define labels that supported the instructor letter grade assignments
Identify High-Error Modules/Steps	Difficulty with high dimensionality of process-level data	Able to identify "high-error" quizzes. Useful for suggesting intervention topic but not useful for prediction approaches
Improved Prediction Models	Simple regression methods (linear regression) preferred to complex methods, difficulty for classification methods	For both regression and classification, simpler methods (linear and logistic regression) preferred to complex methods
Assess Trainee Evaluation Methods	Process-level information was an important contributor to prediction accuracy, suggests the collection of process-level information in rule-based CBT	Summative metrics were more informative than process-level metrics, process-level metrics more useful for TI
Feature Selection	Dimensionality reduction (PCA) required due to high dimensionality of process-level data. Reduced features did not improve clustering performance	PCA and cluster-based dimensionality reduction did not provide as strong prediction performance as full process-level features for both post-hoc and temporal analyses
Inform Intervention Timing / Content	Process-level metrics not able to improve early prediction, related to the shortened training program	Process-level metrics useful for improving early prediction accuracy, "high-error" quizzes provide recommendations of intervention topic

6.2 Implications for Use by Training Evaluators

These results suggest a number of implications for the use of machine learning algorithms in training domains. A specific list is provided here, for use by training evaluators and future researchers:

6.2.1 Training Data

The nature of the dataset has been shown to have major impacts on the machine learning algorithm preparation, selection, and results. It is important for the evaluator to understand the type of training (skill-based, rule-based, knowledge-based), as well as the structure and assessment metrics used in the training program. This aids not only in the preparation and selection of machine learning algorithms, but is critical to the interpretation of the outputs of the algorithms. Additionally, in the analyses presented in this thesis, data compatibility was an important factor in the selection of machine learning models.

6.2.2 Data Preparation

Prior to the use of machine learning techniques, there are several preparatory steps that may be necessary. These include data transformation, selection of a distance measure (for cluster analysis), and feature selection (including dimensionality reduction).

Data Transformation

On the training datasets used in the analysis in this thesis, the variations of data types (subjective vs. objective, summative vs. process level, continuous vs. discrete, etc) encouraged the use of the more flexible range transformation. On both datasets, the cluster performance metrics of compactness and isolation supported the use of the range transformation over the z-score transformation. This is likely due to the variability in assessment metric types in both training environments analyzed here. It is anticipated that datasets for most training programs will exhibit similar properties and thus the use of the range transformation will likely extend to these datasets.

Distance Measure

The increased number of features available in the process-level information provided by CBT and online training programs make the properties of the Euclidean distance measure a good option for these training datasets. The nature of the Manhattan and Chebychev distances tend to result in the process-level metrics dominating the distance measurements in the feature space. The use of weighting may also be necessary to temper the impact

of process-level information on distance measurements, and can either be selected based on inherent relationships in the data (such as the grade contribution for the classroom dataset) or can be informed by subject matter experts (SMEs).

Feature Selection

As modern training programs begin to collect greater and greater numbers of assessment features, it becomes critical to attempt to identify particularly useful features and remove less-informative features. Several potential methods were demonstrated in the analyses in these methods, including PCA, a cluster-based k-means approach, and agglomerative techniques. These methods help to prevent overfitting in supervised analysis when using the many features available in the process-level data, and will become more important with the increased use of CBT and online training programs.

6.2.3 Clustering

In cluster analyses on both datasets presented here, cluster performance favored the use of simple clustering methods (k-means and agglomerative clustering) based on the external metrics of entropy and purity, and the internal metrics of compactness and isolation. As larger datasets with higher numbers of trainees become available, it may be worth revisiting more complex clustering methods, but on datasets similar to those presented here (<50 trainees) the simple clustering methods are better able to capture the relationships in the data.

6.2.4 Supervised Model Selection

In the majority of cases, supervised learning results also supported the use of simple models on these datasets. The findings generally support the use of simple techniques such as linear and logistic regression over more complex methods such as SVMs and ANNs on training datasets such as those collected here. While the relationships within the collected datasets are best supported by simple methods, this result may not extend to larger and more complex datasets that become available in the future through increased use of computer-based environments.

6.2.5 Regression vs. Classification

Careful consideration must be given to the use of regression vs. classification approaches for trainee performance prediction. This primarily relates to the selection of the target variables, and it is important for the evaluator to ensure that the target variable is appropriate and useful as a performance metric for evaluation. There may be cases where a discrete variable is acceptable (such as "A","B","C" grade or "good" vs. "poor"), but other situations where a finer gradation of performance is needed. The selection of the target variable must be carefully considered based on the needs of the training program and the usefulness to training evaluation. The domain was also shown to have an impact on the comparison between regression and classification results, and rule-based settings such as the one examined in this thesis may be more suited to regression rather than classification approaches.

6.2.6 Process-Level Information

There were important advantages provided by the availability of process-level information through the use of CBT and online training in both datasets presented here. However, the advantages differed in each domain, providing improved quality control through post-hoc analyses on the rule-based dataset and potential improved TI timing on the knowledge-based dataset. The results indicate that the length of the training program may have a strong impact on these differences in utility of process-level information. Despite the lack of a consistent effect, the results suggest that training program designers and evaluators should incorporate process-level assessment into the training program in computer-based environments.

6.2.7 Interpretation

In all results presented in this thesis, it was important to provide interpretation of the machine learning outputs. While these algorithms are powerful and will produce a model given almost any dataset, the meaning behind the relationships described in the model and any actions that need to be taken (such as TI or redesigning elements of the program) require the insights provided by a human evaluator. As such,while it is important to recognize the machine learning algorithms can act as a tool to assist in training evaluation,

but should not replace the evaluator.

These lessons can help inform the use of machine learning algorithms in future training evaluation, and ensure that the creation, use, and interpretation of these methods are appropriate for the training domain.

6.3 Limitations

There are several inherent limitations in the results presented in this thesis. First and foremost, the results are limited by the datasets used in the analyses in Chapters 4 and 5. These datasets were collected to be representative of rule-based and knowledge-based training settings, respectively. The results discussed here by definition apply to training datasets similar to those collected, and may not apply to datasets collected under markedly different training program structures or styles. In particular, a skill-based training dataset was not tested in this thesis, and the results presented here cannot be assumed to apply to skill-based training without additional validation.

Additionally, due to logistical considerations, the rule-based dataset collected here only included three training modules, in addition to a test module. This represents a shorter training program than many rule-based training environments currently utilize. Due to the limited number of modules, it is difficult to extract more definite conclusions regarding the temporal predictions and timing of TI on this dataset.

Another limitation relates to the selection of machine learning techniques used in the work. In machine learning as a field, there are countless algorithms and variations of algorithms. The analysis presented in this thesis attempted to select a variety of some of the most commonly used machine learning methods for testing on training data. It is possible that other algorithms or strategies (e.g. decision trees, nearest neighbors, bayesian statistics, boosting, ensembles, etc.) could achieve different or stronger performance results than the algorithms tested here. Additional testing on these and other datasets could investigate the effectiveness of additional algorithms in training evaluation settings.

A final limitation relates to the requirements of the machine learning algorithms, particularly for supervised learning approaches. As discussed in Chapter2, supervised learning algorithms utilize a known dataset including both features and labels to train the

model ("training data") which allow the model to make predictions on previously unseen data ("test data"). This means that in order to use the methods as presented here, the organization must gather a preliminary dataset with both features and labels to train the models before they can be used for prediction. Additionally, these algorithms typically assume the use of the same features across all datapoints. Therefore, any changes to a training program that alter the assessment metrics collected mean that a new dataset must be collected, and old data with different features cannot be combined with this new data to train the models. Thus, these methods cannot be used in a prospective sense to investigate the usefulness of novel assessment metrics or previously untested changes to the training program. Rather, these methods as discussed are only applicable in an established training program that consistently utilizes the same assessment metrics.

6.4 Chapter Summary

This chapter combined the results from Chapters 4 and 5 to extract a set of useful findings and lessons for the application of machine learning algorithms in a variety of training domains. It addressed the potential applications for machine learning in training evaluation discussed in Chapter 3, utilizing evidence from the rule-based and knowledge-based datasets collected. Finally, it identified limitations for the work, based both on the datasets and the methods utilized in this thesis. These elements lay the groundwork for the use of machine learning for training evaluation, and identify the strengths and challenges of machine learning approaches on typical training datasets. The final chapter of this thesis, Chapter 7, presents the conclusions of this work. It discusses the impact and contributions made by this thesis, and potential areas of future work to extend the findings and mitigate some of the limitations identified in this chapter.

Chapter 7

Conclusions

This thesis has presented the considerations (both theoretical and applied) for the use of machine learning algorithms to inform human training evaluation. Chapter 1 introduced the motivation behind the application of machine learning for informing training evaluation, particularly its ability to utilize new process-level information resulting from the increased use of online and computer-based training (CBT) methodologies. Chapter 2 presented background on training and training evaluation, machine learning, and educational data mining (EDM) to frame the discussion of how data types from training programs may impact machine learning approaches. Chapter 3 then discussed relevant factors based on the types of datasets that can arise from training, and the necessary changes to machine learning approaches on these datasets. Chapter 4 presented the first of two example datasets for machine learning testing: a rule-based nuclear reactor CBT. This chapter described the methodology, results, and implications for the use of machine learning in rule-based environments. Chapter 5 presented the second example dataset, from a knowledge-based classroom training environment. It provided a similar discussion of the results of the application of machine learning to this dataset. Chapter 6 compared and contrasted the results from the two environments, provided a set of recommendations for training evaluators on the use of machine learning, and discussed the limitations of the work. This chapter discusses the implications of this thesis, outlining potential areas of future work and finishing with the major contributions of the work.

7.1 Machine Learning In Computer-Based Training Environments

The landscape of modern training has been rapidly changing with the introduction of new technologies. Online and computer-based training (CBT) have begun to make training more flexible, adaptive, and scalable. New training programs are able to be conducted across the world and with far greater numbers of trainees at once. Some of the benefits of computer-based technologies have already been illustrated by the recent popularity of massive open online courses (MOOCs), and organizations have recognized the applications of these technologies in their training programs. However, along with the benefits of CBT programs come unique challenges posed by the nature of these technologies. In particular, training programs are able to generate vast amounts of data, both in terms of the number of trainees and the higher specificity of the assessment metrics that can be gathered. Supervisors and evaluators of training programs are entering the world of "big data" where traditional analysis techniques and human intuition have difficulty in drawing meaning from the data.

To address this issue, this thesis has proposed the use of machine learning techniques. With the rise of large datasets in other fields, machine learning has become one of the primary tools for analyzing these datasets in an efficient and meaningful way. However, the nature of human training data is unique from the typical machine learning domains, and it is important to understand both the potential applications of machine learning to training as well as the implications of training data on the algorithms. This thesis first introduced background information on typical training domains and on machine learning techniques. In particular, a taxonomy of training domains was developed based on whether the target training tasks fell under Skill, Rule, or Knowledge (SRK) domains. Each of these domains contains properties specific to that domain in the typical training methods and assessment data collected. The problem of applying machine learning to training datasets was broken down into addressing the applications to each of these domains.

Chapter 3 then investigated the impact of the properties of training data from each of these domains on machine learning approaches. It identified challenges that arise in the limited datasets, feature set size, and noise properties of training data that differ

from traditional machine learning datasets. There are also several considerations that must be taken into account prior to the use of machine learning on training data, such as data standardization, selection of a distance measure in the feature space, the particular algorithms to be tested, and the metrics by which algorithm performance can be compared. A set of representative machine learning algorithms of varying complexity was selected for both unsupervised and supervised learning approaches. Unsupervised algorithms included k-means, hierarchical clustering, Gaussian Mixture Models (GMMs), and Self-Organizing Maps (SOMs). The supervised algorithms tested were linear regression, logistic regression, Support Vector Machine (SVMs) and Artificial Neural Networks (ANNs). These were selected as a range of some of the most common machine learning techniques and spanning a range of complexity in implementation and interpretation. A set of potential applications of machine learning to the training domain were also identified, making use of unsupervised algorithms in a descriptive fashion and supervised algorithms as a predictive approach.

Due to the differences in datasets across different domains in the SRK framework, example datasets were collected for both rule-based and knowledge-based environments to test these applications. The analysis focused on these domains due to the greater use of CBT methods and higher cognitive complexity as compared to skill-based domains. The rule-based dataset used was gathered from a simulated nuclear reactor CBT environment and was dominated by the use of procedures that are common in rule-based settings. The knowledge-based dataset was taken from a collegiate course that included online elements in addition to traditional classroom assessment techniques such as tests, projects, and problem sets. Analysis of machine learning on these datasets allowed not only the determination of whether the potential applications could be realized, but also the comparison of machine learning results across training domains.

The analysis of these datasets, which were presented in Chapters 4 and 5, revealed several important findings about the use of machine learning on training datasets. First, an important challenge arose in both datasets relating to limited size of the datasets. While computer-based training technologies allowed for the collection of much greater detailed assessments than were available through traditional methods, the number of trainees are small compared to traditional machine learning settings. Thus, the use of dimensionality reduction techniques was critical in reducing the feature space and preventing overfitting

of the data. In the future, larger and larger datasets may become available as organizations take advantage of the benefits in scalability of CBT technologies, such as seen in massive online open courses (MOOCs). However, most current training programs will encounter limitations of dataset size, and this work has demonstrated the effectiveness of several types of dimensionality reduction, including clustering techniques and principal component analysis (PCA). This will be an important element of future applications of machine learning approaches to training datasets.

Another important finding of this work was that in both datasets simpler machine learning techniques (such as k-means, linear and logistic regression) generally performed as well or better than more complex methods (such as SOMs, SVMs, and ANNs). These simpler algorithms offer additional advantages in ease of interpretability compared to more complex models. When utilizing machine learning techniques to inform training evaluation, it is important for the evaluator to be able to understand the relationships between the inputs and the outputs of the model. This can be important from a programmatic perspective such as identifying inefficient assessment metrics to be removed in future iterations of the training program, or from a trainee assessment perspective such as determining the appropriate timing for trainee intervention (TI).

A significant advantage of using computers in training is the ability to efficiently collect process-level information in addition to traditional summative assessment metrics. The collected datasets allowed for the comparison of the contributions of each of these assessment types to model performance. In general, it was found that process-level metrics are useful when summative metrics are not available. This was seen in the rule-based setting where the truncated training program allowed for process-level metrics to make contributions to post-hoc prediction. It was also observed in the knowledge-based dataset where the process-level information improved the accuracy for predictions early in the training program when summative metrics were not yet available. The ability of the process-level information to improve the accuracy of early predictions has important implications on TI. When selecting the timing and trainees for intervention, it is useful not only to have accurate predictions of later performance, but to have these predictions earlier in the training program. Through early identification of trainees in need of intervention, the training program can save money and improve trainee success rate.

A recurring theme throughout this research was the importance of human interpre-

tation of the model outputs. While the machine learning techniques can allow for the greater description and prediction of trainee performance, the decision of how to use this information still must be made by a human. These algorithms are not a replacement for training supervisors; they instead act as a tool that can improve the efficiency and effectiveness of training evaluation as training programs adapt to utilize computer-based elements.

7.2 Future Work

This work has provided the foundation for the study of machine learning techniques in training, but is by no means exhausts the possibilities of research in this area. This section outlines some extensions of this work that could provide even greater benefits for training evaluation.

7.2.1 New Datasets

Obtaining appropriate datasets is a major challenge to research in this area. Training data are often proprietary, as it relates to performance of the trainee in the operational environment. Consider the example of nuclear reactor operator training. Both the organizations and trainees typically do not want their training data to be made public, due to the concern that errors contained in the data could be used as evidence against the organization or the trainee. Therefore, obtaining large, real-world datasets can be challenging, which leaves considerable future work that can be done with additional datasets.

Thus, an important area of extension includes the collection of new and larger datasets. As training programs scale to include much greater numbers of trainees by utilizing computer-based technologies, new datasets can be collected that mitigate some of the challenges encountered in the datasets presented in this thesis. Specifically, larger datasets could reduce the need for dimensionality reduction and could exhibit additional complexities that algorithms such as ANNs could exploit.

Additionally, the present work did not analyze a skill-based training dataset. As technologies such as video capture systems in sports become more developed and are used in training programs for athletes, it may become possible to utilize machine learning on these datasets as well. The burgeoning field of sports statistics is already recognizing

the potential for new technologies to generate vast datasets of player movement and performance (an example from basketball is shown in Figure 7-1), and machine learning may play an important role in the analysis of these datasets.

7.2.2 Novel Evaluation Metrics

The nature of the assessment metrics collected as part of a training program has been shown to strongly impact the usefulness of machine learning methods on those features. The development of metrics that are better at assessing performance and learning may dramatically improve the ability of machine learning methods to aid in training evaluation. Thus, research on novel metrics closely parallels the continued study of machine learning techniques on training data. The rule-based dataset presented in this thesis was particularly challenging for machine learning algorithms, which could have been a result of poor assessment metrics. Better measurements of procedure adherence as well as

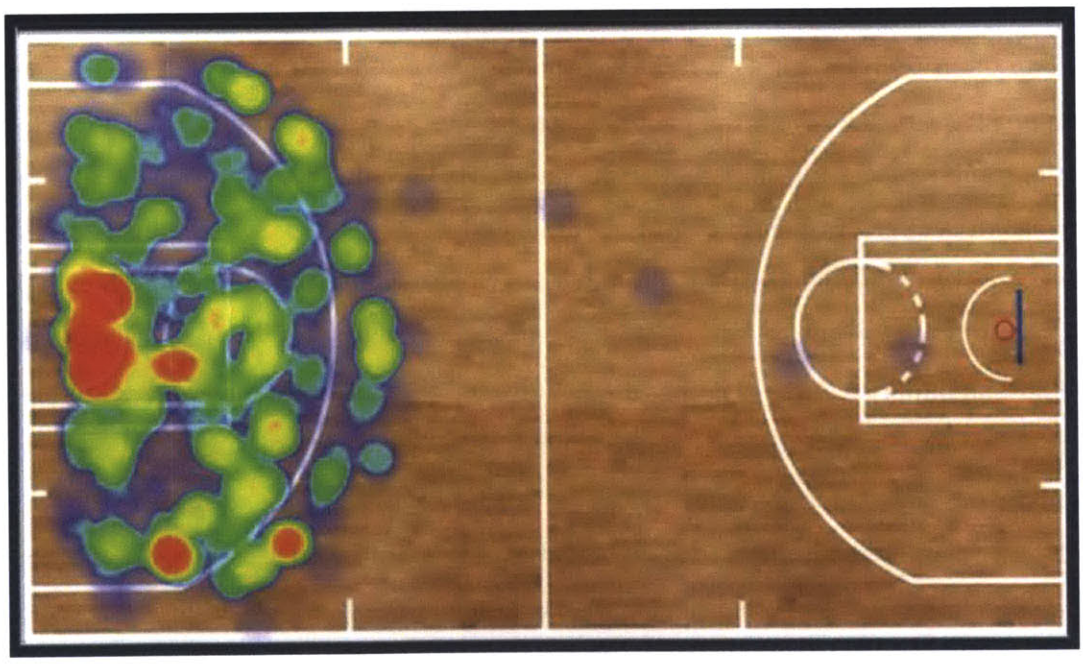


Figure 7-1: Example positional data from the SportVU player optical tracking system. Reproduced with permission from STATS LLC.

greater use of system performance metrics could improve the prediciton accuracies over those achieved in Chapter 4. As new metrics become available, it will likely be worth revisiting the algorithms tested in this work on the new feature types.

7.2.3 Additional Algorithms and Strategies

The field of machine learning has developed a vast set of tools for analyzing large and complex datasets, of which the algorithms tested here comprise only a small subset. Future research could utilize analyses similar to those presented here to investigate different machine learning algorithms or strategies. For example, Bayesian statistics focus on a probabilistic approach to machine learning, which could be useful in addressing the noise associated with human performance data [12]. Decision trees are another common simple machine learning method, and the advantages of simple models on training data shown in this thesis imply that decision trees could be a successful technique. Additionally, the strength of simple methods on the presented datasets also implies that strategies such as ensembles or boosting that utilize multiple simple models to improve prediction may have success on similar training datasets. Other more complex models such as "deep learning" would likely struggle on the datasets presented here, but could take advantage of larger datasets as they become available.

Additionally, feature weighting in unsupervised learning could improve cluster algorithm performance if some features are known to be more important than others. The importance of particular features or trainee behaviors could be provided by subject matter experts (SMEs) to inform weighting strategies during clustering. Further investigations could be conducted to analyze the usefulness of subjective weightings provided by SMEs in unsupervised algorithm performance on training data.

7.2.4 Interface Design

This thesis has presented the potential applications of machine learning algorithms to training datasets for the purposes of assisting training evaluators. In a real-world setting, it would be useful to training evaluators for these algorithms to be wrapped in a user-friendly interface that allows for the easy input of data and clear presentation of results. Future research could investigate constructing a user interface tool for training supervisors that implement the algorithms discussed in this work.

7.2.5 Datasets with Missing Features

As discussed in Chapter 6, an important limitation of the work presented in this thesis is that the methods are restricted to cases where the same assessment metrics are consistently collected. If the training program frequently changes in terms of modules, learning material, or metrics, it is difficult to create a dataset sufficient to train the machine learning models. Specifically, if the measurement types used as features in the model vary, a model created on all of the possible features will have missing data for some features. Several data preprocessing methods that have been developed to handle datasets with missing information which could be utilized to address combining datasets with different features (e.g. [153]). These along with semi-supervised learning techniques could allow the methods presented in this work to be expanded to cases where changes in training program create missing information in the dataset [154]. These strategies could be investigated in future research to extend the findings in this thesis to training programs that frequently modify the assessment metrics used.

7.2.6 Reinforcement Learning

Machine learning also offers an entirely different strategy that could be useful to training evaluation, termed reinforcement learning. These algorithms do not focus on description or prediction as the unsupervised and supervised algorithms presented in this work, but instead construct models that attempt to take optimal actions given the current state of the system. These algorithms are commonly used in robotics or path planning, but could be applied to training as well. In training, reinforcement learning could be used to develop models that do not analyze what the trainee *does*, but rather indicate what they *should* do. In this way, these models could act similarly to an expert trainee (when a human is not available), and allow for comparison between a trainee's performance and the "optimal" performance as indicated by the model. Additionally, these models could be used to inform procedure design by identifying shorter or alternate paths through the procedure.

7.2.7 Models of Learning

The research presented in this thesis focused on describing and predicting trainee performance (also called "training transfer" in training evaluation literature). That is, it utilized the actions that the trainees actually performed in an attempt to predict actions or mistakes that might occur in the future. This performance-level modeling ignores the underlying cognitive states of the trainee, and does not directly assess the learning that takes place. While the performance in the operational environment is of paramount importance, it may be of interest to researchers to link these results with the cognitive processes of the trainees. These resulting models could be particularly important for knowledge-based training settings, where the development of abstract representations and mental models by the trainees is an important part of the training program.

7.3 Contributions

The overall objective of this thesis was to identify and test the theoretical considerations and applications of machine learning algorithms to assist with training evaluation. In working towards this objective, several contributions were made to the fields of training, training evaluation, and machine learning on human behavior. This thesis acts as a foundation for the research of machine learning on training datasets, and provides important lessons for both future researchers and practitioners in this area. The contributions include both theoretical considerations for the use of machine learning in training, as well as findings from the testing on rule-based and knowledge-based datasets.

Chapter 1 presented three specific research goals to be covered by this thesis. The first goal was to "determine the typical properties of CBT data, how the training domain impacts the data obtained in these settings, and the considerations of applying machine learning algorithms to training data." In addressing this goal, this thesis makes the following contributions:

- The development of a taxonomy of training domains based on the training task: Skill-based, Rule-based, and Knowledge-based. In this taxonomy, there are strong commonalities in the training methods and assessment techniques within a particular domain. This taxonomy was instrumental in this research for identifying

different properties of training data and could be a useful taxonomy to compare training domains for other fields.

- CBT methods allow for the collection of action-by-action adherence information in rule-based domains. This thesis provided a discussion of potential measurement techniques for procedure adherence. It proposed treating a procedure as a sequence, enabling the use of sequence distance techniques for the comparison of the trainee actions to the prescribed actions in the procedure. Additionally, this work described the creation of a model-based Procedure Adherence Metric (PAM) for the measurement of procedure adherence in rule-based environments. Computationally, simpler sequence distance metrics (such as Levenshtein distance) out-performed the PAM in machine learning analyses, but the PAM offers advantages in sensitivity to action ordering that may merit further investigation of model-based techniques for measuring procedure adherence in rule-based domains.

- The identification of necessary preparations on training datasets prior to use in machine learning approaches. The variation in training data types and limited dataset sizes must be addressed prior to use in machine learning. Data standardization, selection of a distance measure, and dimensionality reduction must all be considered prior to the application of unsupervised and supervised approaches.

- Due to the importance of dimensionality reduction on these datasets, this thesis introduced several potential methods for reducing the dimensionality of the feature space. For post-hoc analyses, aggregate metrics could be used, such as averaging over quizzes or other process-level metrics. Cluster-based techniques were also tested, as demonstrated by the separation of quizzes in the knowledge-based dataset into "high-error" and "low-error" categories. PCA was also used as a traditional machine learning dimensionality reduction technique. Both could have utility to training supervisors: PCA methods demonstrated benefits to supervised prediction accuracy, while cluster-based strategies were able to suggest potential topics for TI.

The second goal presented in Chapter 1 was to "assess the usefulness of supervised and unsupervised learning algorithms in example training sets from different types of training domains, particularly from the perspective of a training evaluator." A set of potential applications of machine learning to training were presented in Chapter 3, and

these were demonstrated on both rule-based and knowledge-based datasets. This thesis makes the following contributions towards this goal:

- Clustering approaches were demonstrated to be able to inform labeling for discrete targets. In knowledge-based datasets, a letter grade is commonly used in addition to a numeric grade and often is subjectively determined by the training supervisor. In the example knowledge-based dataset collected, it was shown that clustering was able to objectively identify natural splits in the data for grade assignments. This could assist training supervisors in determining where to set grade cutoffs. When combined with supervised learning that predicts trainees in need of intervention, the cluster results could be used to select trainees with similar performance for combined intervention.

- In post-hoc prediction analyses, it was demonstrated that supervised learning algorithms could indicate to a training designer as to which assessment metrics provide the highest predictive capacity. Since a principal goal of training assessment metrics is to be predictive of operational performance, this information could be used to inform future iterations of the training program design, by focusing on metrics that are predictive of future performance and eliminating inefficient metrics.

- Supervised learning algorithms were shown to be able to achieve accurate predictions of future trainee performance early in a training program. A temporal analysis of overall prediction accuracy indicated that trainees could be selected for intervention prior to important assessment landmarks in the training program (such as before the first test in the knowledge-based dataset). The predictions were improved by the inclusion of process-level information, discussed further below. Overall, these predictions have great value to training supervisors, as the earlier an intervention is applied the greater the potential effect on performance and cost savings.

The third goal introduced in Chapter 1 was to "determine the importance of the detailed temporal and process-level information provided by online and CBT formats in the application of machine learning to training data." As previously mentioned, an important advantage to the use of computer-based technologies in training is the opportunity to gather assessment metrics at a more detailed level of data specificity, such as process-level data. In addressing this goal, the following contributions were made:

- This work resulted in an improved understanding of the relative contributions of summative and process-level assessment metrics to prediction performance. Specifically, summative metrics were shown to provide greater prediction performance than process-level metrics on both rule-based and knowledge-based datasets. Thus, when available these metrics dominate the prediction performance and thus are a critical part of the training program. However, when summative information is limited typically earlier in a training program, process-level information has the potential to significantly contribute to prediction performance.

- Process-level information was shown to have the potential to improve early predictions of final performance over summative metrics alone. Since process-level information is collected much more frequently than summative metrics, it was shown that predictions could be made earlier and with greater accuracy by utilizing process-level information (which is important for TI). Thus, this work suggests that the collection of these process level metrics should be incorporated into future training programs.

This dissertation provides the groundwork for the use of machine learning algorithms to inform training evaluation in modern training environments. There are two primary populations to which this work applies. The first are researchers, from both training evaluation and machine learning fields. For training evaluation researchers, this work suggests the basis for how machine learning algorithms can improve the success of training programs and the trainees that come out of those programs. For machine learning researchers, the analyses contained in this work comprise an interesting example of the use of machine learning on datasets representing human behavior. Future research attempting to use machine learning models on humans in other settings can make use of the findings presented here. The second group are training practitioners, for whom this work represents a framework for the implementation of these algorithms in their own domains. Following the lessons and recommendations contained here could mitigate the challenges these practitioners encounter in the introduction of these methods in their programs.

Training programs exist in almost every organization, and while useful can be expensive and time consuming to implement. As modern training programs incorporate computer-based elements, it is important for these programs to be able to take advantage of improvements in scalability and data collection provided by these elements. These

benefits provide new opportunities for training evaluation, which can improve the efficiency, efficacy, and cost-effectiveness of the programs. This dissertation has provided one method for utilizing the larger, more complex datasets arising from these new training environments: machine learning. In doing so, it has provided a set of lessons for the future use of these algorithms on human training data and paved the way for future research in this area. Future directions could focus on the use of new datasets or alternative algorithms that build off of the foundations in this thesis. This dissertation and future work should pave the way for improving the efficiency and cost effectivness of training programs that will reach wider audiences and train the next generation of professionals.

Appendix A

LOFT Scenario

This appendix presents the details of the LOFT scenario discussed in Chapter 3. This scenario is reproduced from [142]. The scenario itself is first presented starting on the following page, followed by the decriptions of the problems to be inserted into the scenario.

PAN AM LOFT SCENARIO (9-26-88)

CLIPPER 594 "HEAVY" IAD-JFK (A-310)

Problems 1, 5, 6, 7 (See problem menu)

1) SIM setup		Dulles runway 01R (#), Gate #3, taxi weight 233 900 lb, fuel 22 500 lb, take-off CG 29.2%, ceiling 1 000 ft, cloud tops 3 000 ft, visibility 10 000 RVR, OAT 30F (-2C), altimeter 29.59 Hg (1 002 mb), wind 020/8, QXI/OCI #1: Green-to-blue hydraulic PTU INOP. QXI/OCI #2: Left inner fuel tank pump 1 INOP. **Insert Problem 1.**
2) Dep ATIS	134.85	*"This is Washington Dulles departure information ZULU. Ceiling measured 900 overcast, visibility 2 miles in light snow, temperature 30, dew point 28, wind 020 at 8, altimeter 29.59. Departures expect runway 01 right. Inform clearance or ground control on initial contact that you have received information ZULU."*
3) Clearance delivery	127.35	*"Clipper 594 "Heavy", cleared to JFK, capital two departure as filed, maintain 4 000 ft, expect 17 000 ft ten minutes after take-off. Departure control frequency is 125.05, squawk 0523, contact ramp control on 129.55 prior to taxi."*
4) Routing		Radar vectors direct Baltimore, V-44, V-229 MORTN, V-44 CAMRN, direct JFK.
5) Ground support		Clearance to pressurize hydraulics, remove external electric (as appropriate). Clearance to start engines when requested. Remove external connections when directed. *"Standby for hand signals on your left."*
6) Ramp control	129.55	Receive pushback request. *"Clipper 594 "Heavy", cleared to push back, face east."* Receive taxi request. *"Clipper 594 "Heavy", taxi eastbound to taxiway Echo-1, turn right and taxi south, then contact Dulles ground control frequency 121.9."*
7) Ground control	121.9	*"Clipper 594 "Heavy", continue taxi and hold short of runway 01 right."*
8) Atlanta flight support	130.9	Receive blocks departure message.
9) PANOPS	129.7	Receive off blocks time and gallons of fuel added.
10) Load control	129.7	*"Clipper 594 "Heavy", load control. Your zero fuel weight is 210.6 with a CG of 27.2; your take-off weight is 233.1 with a CG of 29.2. Passenger load is 12 first class, 21 clipper, and 103 coach. Stabilizer setting is 0.1 up."*
11) Ground control	121.9	(Approaching runway 01R) *"Clipper 594 "Heavy", contact Dulles tower, frequency 120.1."*
12) Tower	120.1	*"Clipper 594 "Heavy", wind 330 at 15 maintain runway heading, cleared for take-off."*
13) Tower	120.1	*"Clipper 594 "Heavy", turn right heading 080, vectors on course, contact departure control frequency 125.05."*

14) Departure control	125.05	*"Clipper 594 "Heavy", radar contact, continue heading 080, vectors to Baltimore, climb to and maintain 6 000 ft, receiving Baltimore cleared direct."*
15) Departure control	125.05	(Approximately 20 miles west of Baltimore VOR) *"Clipper 594 "Heavy", continue climb, maintain 17 000 ft, contact Washington Centre on 133.9."*
16) Washington Centre	133.9	*"Clipper 594 "Heavy", radar contact, maintain 17 000 ft and cleared via flight plan route."*
17) Atlanta flight support	131.25	Receive airborne message.
18) Washington Centre	133.9	(Approximately 41 miles west of Sea Isle) *"Clipper 594 "Heavy", contact Washington Centre on 127.7."*
19) Washington Centre	127.7	*"Clipper 594 "Heavy", radar contact, maintain 17 000 ft."*
20) ARVL ATIS	115.4"	*"This is Kennedy International Airport information WHISKEY. Sky condition 800 overcast, visibility 1 and 1/4 mile in snow. Temperature 29, dew point 27, wind 310 at 3 knots, altimeter 29.75. Arrivals expect VOR/DME approach runway 22L. Notice to airmen, ILS 22L out of service. Departures expect runway 22R. Inform New York approach control on initial contact that you have received Kennedy arrival information WHISKEY."*
21) Washington Centre	127.7	(Overhead Atlantic City) *"Clipper 594 "Heavy", descend and maintain 10 000 ft, Kennedy altimeter 29.75 Hg (1 007.5 mb)."*
22) Washington Centre	127.7	(5 miles northeast of Atlantic City) *"Clipper 594 "Heavy", contact New York Centre on 128.3."*
23) New York Centre	128.3	*"Clipper 594 "Heavy", radar contact, maintain 10 000 ft, cleared CAMRN one arrival JFK."*
24) SIM setup		JFK runway 22L (#), ceiling 800 ft, cloud tops 6 000 ft, visibility 8 000 RVR, temperature 29F (-6C), altimeter 29.75 Hg (1 007.5 mb), wind 210/04.
25) Problem		(10 miles northeast of Atlantic City) **Insert Problem 5 or 6 or 7.**
26) PANOPS	131.37	(Receive in-range message) *"Clipper 594 "Heavy", you can expect gate number 3, enter via taxiway KILO."* Provide assistance as requested.
27) New York Centre	128.3	(5 miles southwest of CAMRN) *"Clipper 594 "Heavy", contact New York approach control on frequency 127.4."*
28) New York approach control	127.4	*Clipper 594 "Heavy", radar contact, fly heading 040 and descend to 3 000 ft. Vectors for the VOR final approach course runway 22 left."* (on final vector) *"Clipper 594 "Heavy", cleared for the approach, contact Kennedy tower on frequency 119.1."*
29) Kennedy tower	119.1	*"Clipper 594 "Heavy", wind 210 at 4 knots, cleared to land on runway 22 left."*

30) Kennedy tower	119.1	(During rollout) *"Clipper 594 "Heavy", turn right first available taxiway, hold short of runway 22 right, remain this frequency."*
31) PANOPS	131.37	Provide assistance as requested.
32) Kennedy tower	119.1	(Approaching runway 04 left) *"Clipper 594 "Heavy", cross runway 22 right, left on the inner, contact Kennedy ground control on frequency 121.9."*
33) Kennedy ground	121.9	*"Clipper 594 "Heavy", taxi via the inner to your gate."*
34) Atlanta flight support	131.25	Receive blocks arrival message.

LOFT profile codes: LFT = Normal route between airports

LRR = Abnormal route between airports

LTB = Turnback or diversion

Alternate Weather Reports (If Requested)

Newark: 300 obscured. Visibility 1/2 mile, snow, fog. Temperature 30, dew point 29, wind 350 at 5 knots, altimeter 29.72.

Philadelphia: 400 obscured. Visibility 1/2 mile, snow, fog. Temperature 31, dew point 29, wind 010 at 4 knots, altimeter 29.70.

Boston: Measured 800 overcast. Visibility 3 miles, snow. Temperature 15, dew point 11, wind 010 at 7 knots, altimeter 29.58.

Bradley: Measured 400 overcast. Visibility 3/4 mile, snow. Temperature 20, dew point 17, wind 020 at 5 knots, altimeter 29.68.

Baltimore: Estimated 400 overcast. Visibility 1 mile, snow, fog. Temperature 30, dew point 27, wind 020 at 7 knots, altimeter 29.59.

Andrews AFB: Measured 400 overcast. Visibility 1 mile, snow. Temperature 31, dew point 27, wind 020 at 5 knots. Altimeter 29.60.

PAN AM LOFT PROBLEM MENU (A-310)

REVISED (9-26-88)

Problems and/or situations

1 Engine potential hot start

2 Engine stall

EGT exceeds 644 degrees

Engine shut down

3 Engine oil low pressure

Engine shutdown

4 Green hydraulic system failed

5 Bravo Whiskey Direct

"Clipper 594, New York, contact your company immediately on frequency_____." (Company frequency)

(When contacted)

"Clipper 594, flight control, we have just been notified by Security of a Bravo Whiskey Direct for your flight. Security has confirmed the threat to be valid. We advise you to land immediately at_____(Planned destination airport)."

Provide assistance as requested.

Provide priority ATC handling.

Any runway available for landing.

6 Passenger threat

Flight attendant reports that a passenger has barricaded himself in an aft lavatory; he claims to have a gasoline bomb device (or hand grenade) which he continually threatens to detonate; he is demanding that the flight divert to _____ (Nicaragua, Beirut, Tehran, etc. as appropriate).

7 Communication failure

Crew loses all communications with air traffic control on normal VHF frequencies; also unable to establish contact on 121.5 or receive on VOR frequencies; maintain loss of communications as long as possible; attempted communications with approach control are successful; instructions are for the flight to *"continue last assigned clearance"*; give holding instructions if requested.

NOTE: Reason for loss of all radios is massive explosion in the air traffic control building.

8 Passenger incapacitation (or intoxication)

Flight attendant reports that certain individual has suffered massive seizure of unknown type (or is extremely unruly and is purposely obstructing cabin crew duties).

9 Brake explosion/green system hydraulic failure

Brakes hot indication (any wheel) followed shortly thereafter by a green system hydraulic failure; flight attendant reports loud noise below floor; possible damage in the wheel well.

10 Suspicious object

Flight attendant finds device in lavatory area which resembles a bomb; device looks like two sticks of dynamite with ticking object attached with tape.

LOFT profile codes: LFT = Normal route between airports

LRR = Abnormal route between airports

LTB = Turnback or diversion

Appendix B

Measuring Procedure Adherence and Development of the Procedure Adherence Metric (PAM)

B.1 Measuring Procedure Adherence

Procedure adherence represents the faithfulness to which the operator follows the prescribed procedures. Prior research on adherence to procedures has primarily been part of studies on human error in rule-based environments [155–157]. This research has focused on the theoretical aspects of errors while following procedures and the modeling of human error in these cases, such as the Memory For Goals (MFG) model [158, 159]. This work identifies differences in tendencies for adherence across cultures and levels of experience, and often utilizes interviews with operators rather than experimental studies (e.g. [155, 160]). A few experimental studies have been conducted, primarily focusing on the nature of errors in procedure-following or improvement of adherence through the design of procedures or new interfaces [161, 162]. There are few sources that discuss the measurement of procedure adherence, and these do not discuss the implications of procedure adherence as an assessment metric in training. Thus, one of the important contributions of this work includes the discussion of potential measures of procedure adherence, and their use both as a training assessment metric and as features in machine learning approaches. These topics are covered in this and the following sections.

Adherence to procedures can have several interpretations dependent upon the partic-

ular domain in which the procedures are used. In nuclear power plant operation, there are often many procedures available contained in either binders or more recently in computerized checklist systems. Thus, an important aspect of adherence in nuclear power plants is the ability to select the correct procedure to be used based on the current system state. This ability is referred to in this appendix as "procedure selection." When an incorrect procedure is selected, subsequent actions performed in the following of that procedure are unlikely to be appropriate for the situation.

Once a procedure is selected, a second type of adherence (simply termed "adherence" here) arises relating to the accurate completion of all appropriate steps contained in that procedure. Sometimes procedures will be branched, in which case not all steps will be used. In these cases, adherence can be measured to following all the steps along one set of sequential actions, i.e. the "path" through the procedure. Sometimes only a single path is correct given the system state; other times there may be multiple "correct" paths for appropriate operation. In the former case, the procedure is linear, and at any time during the course of completing the procedure there is only one intended action for the next step in the procedure. Thus, adherence can be measured based on whether the next action is the intended action. When multiple correct paths exist, adherence is much more difficult to measure, as actions that are incorrect for one path may still be correct along a different path.

There are a variety of metrics that could be used to measure procedure selection and adherence. For procedure selection, the primary information of interest is whether or not the correct procedure was selected. The simplest assessment metric in training could then be a binary of whether the trainee did ("1") or did not ("0") choose the correct procedure. Another option would be to apply a performance penalty to those trainees who did not select the correct procedure, which would typically manifest itself as a weighted case of the binary metric (e.g. "-10 points if they did not select the correct procedure"). In machine learning, any transformation used (such as range or z-score transformations described in Chapter 3) automatically reweights features to an approximately equal scale and would thus account for any weighting applied to the binary metric. Thus, procedure selection in this analysis is assessed by the simple binary metric rather than any weighted form.

In the consideration of adherence, it is useful to think about a procedure as a sequence, defined as an "ordered set of events." An SOP defines a series of actions for the

user to take, typically under a certain set of initial conditions that make the procedure appropriate. The set of actions contained in a procedure can be translated into a sequence, with each action having a previous action and a subsequent action (see Figure Q-1). A trainee similarly generates an ordered sequence of actions as s/he attempts to complete the procedure as part of the training module. In a sequence of actions, common errors include omission of an action, performing actions out of order, or substitution of an action with an incorrect one. These errors create mismatches between the procedure sequence and the trainee sequence. Typically in a complex system, there are more actions available to the user than are needed for any particular procedure. Consider the aircraft training example given in Chapter 3; during the pre-takeoff checklist the pilot is not required to press every button in the cockpit. With more actions available than used in the procedure, it is possible for a trainee sequence to contain actions that are never observed in the procedure sequence.

In this framework, procedure adherence can be measured by the difference between a

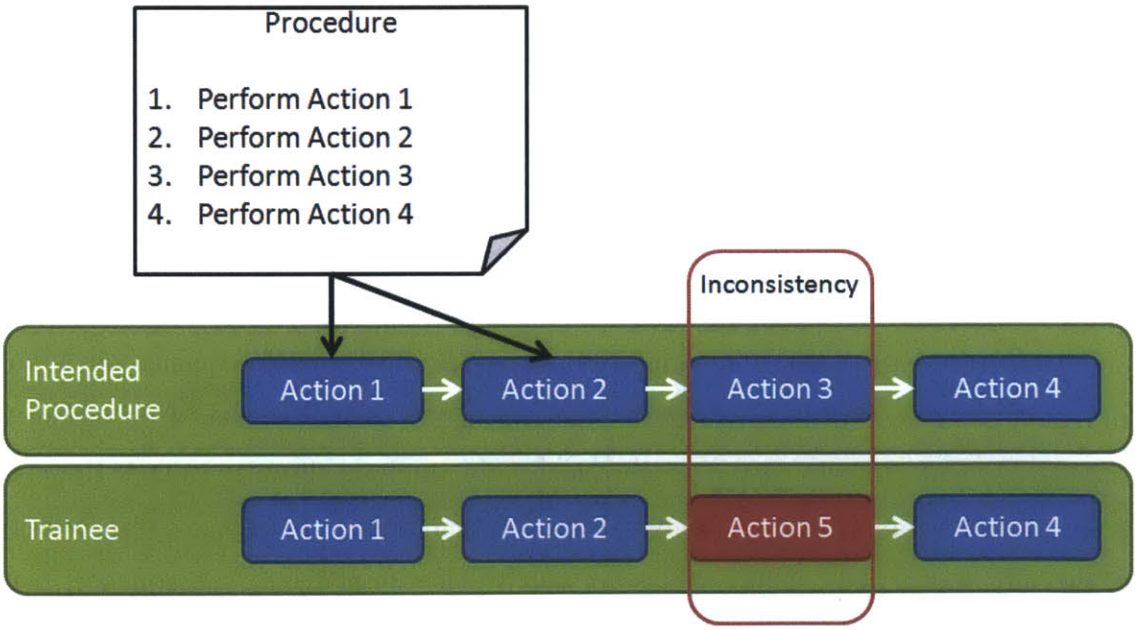


Figure B-1: Viewing a procedure as a sequence.

SOP sequence and the trainee sequence. Numerous methods that measure the distance between sequences have been developed, including sequence-based methods (e.g. Levenshtein distance [163]) and model-based methods (e.g. Kullback-Leibler divergence [164]). Sequence-based methods focus on the direct comparison of the sequences, while model-based methods model each sequence and then compare the similarity of the models as a proxy for sequence distance. To select the best method for the calculation of sequence distance, four elements important to procedure adherence measurement are:

- Ability for different types of errors to be penalized separately (i.e. error weighting)
- Non-equal sequence length between the observed and expected sequence of actions, i.e., where the user performs a different number of actions than the prescribed procedure
- Sensitivity to number of deviations between the observed versus expected sequence of actions
- Sensitivity to trainee action order: the ability to distinguish between correct and incorrect ordering

Three methods that calculate sequence distance were considered as possible features for machine learning approaches to measuring procedure adherence: two sequence-based methods (Levenshtein distance and suffix arrays [165]) and one model-based method (Kullback-Leibler divergence). The details of the calculation of each method and their comparison on the four criteria presented above can be found in Appendix C. Briefly, the Levenshtein distance is calculated by the minimum number of edits (insertions, deletions, or substitutions) to change the trainee sequence into the intended sequence. It is able to handle sequences of unequal lengths and is sensitive to the number of deviations of the trainee, and provides easy interpretation of the resultant distance values. However, it weights all errors equally and only considers action order at the level of transposition of two adjacent actions. Suffix arrays are able to identify matching subsequences beteween the trainee and prescribed sequence, and thus account for the ordering of actions. However, the distance measure focuses on the largest matching subsequence, and thus does not account directly for the number of deviations. The Kullback-Leibler (KL) approach exhibits similar adherence detection properties to the Levenshtein distance, but

additionally accounts for action ordering and weights errors based on the frequencies of actions and transitions encountered in the prescribed sequence. Thus, the KL divergence is able to meet all four criteria specified above.

Two metrics stand out as potential candidates for measuring adherence in training: the Levenshtein distance for its simplicity and ease of interpretation, and the KL divergence by meeting all of the desired criteria. However, the KL divergence requires additional modifications before it can be used directly for measuring procedure adherence. Thus, a new metric was created, termed the Procedure Adherence Metric (PAM), that captures the benefits of using the KL divergence approach but is able to be calculated directly from sequences generated by trainees in rule-based environments. The following section details the calcuation of the PAM.

B.1.1 Procedure Adherence Metric (PAM)

The main goal of measuring procedure adherence is to assess trainees' performance against the SOP. Additionally, trainees can be objectively compared against each other based on their training performance, and tracking procedure adherence can indicate struggling trainees that need re-training. The Procedure Adherence Metric (PAM) was based on the KL divergence between the trainee and intended action sequences. Formally, the KL divergence between two sequences can be calculated as shown in Equation Q.1.

$$D_{KL}(P||Q) = \sum_i ln(\frac{P(i)}{Q(i)})P(i) \tag{B.1}$$

where $P(i)$ is the probability of observing the transition from action $i-1$ to action i in one sequence (such as the intended sequence), and $Q(i)$ is the probability of observing the transition from action $i-1$ to action i in the other sequence (such as the trainee sequence). As can be seen from this formula, the KL divergence requires at least two actions in the sequence, as no transitions exist with sequences containing only one action. Also, the basic form of the KL divergence is not symmetric, meaning that $D_{KL}(P||Q)$ is not necessarily equivalent to $D_{KL}(Q||P)$. For calculating adherence, it is useful to have a metric that does not depend upon which sequence is used for P and Q, and thus a symmetrized form can be used, as shown in Equation Q.2.

$$D_{symmetricKL} = D_{KL}(P||Q) + D_{KL}(Q||P) \tag{B.2}$$

In this symmetrized form, the divergence will be the same regardless of which sequence is used for P and Q. An adherence metric based on this divergence could use a single final value calculated after the trainee has completed the module, or could utilize KL divergence partway through the the training module. Consider a trainee who completes 30 actions during a training module. At any given point during the training module, the "current" KL divergence between the trainee sequence and the intended sequence at that point in the module can be calculated. Thus a sequence of KL divergences can be generated over time, but it is unclear which aspects of this divergence sequence are the most useful for measuring adherence.

KL divergence can be measured in many ways: maximum KL divergence, the mean KL divergence, the final KL divergence (using only the full sequence), and the sum of the KL divergences in the sequence. It was determined that using the sum of the KL divergence values over all actions in a training module provided the closest rankings of trainees (from best to worst) as compared to an expert evaluator (Appendix D). Using the sum of KL divergences as the basis for the adherence metric, the PAM value can then be calculated as shown in Equation Q.3.

$$PAM = \sum_{i=1}^{N} D_{KL} \tag{B.3}$$

where N is the number of events or actions in the training sequence, and D_{KL} represents the symmetrized Kullback-Leibler divergence between the trainee sequence of states $1i$ and the intended sequence of states of equivalent length. If N is greater than the number of states in the intended sequence (M), the complete intended sequence is used for all $i > M$. It is important to recognize that as the PAM is based on divergence, and a lower score indicates better performance.

A potential issue that arises in the use of the KL divergence for the PAM is zero-probability values in the transition matrix. This transition matrix represents the probability of all transitions between states in the model. For a model based on a sequence, the maximum likelihood estimate simply counts the number of times a consecutive action pair is found (e.g. action 1 to action 2), and normalizes by the number of transitions.

An example based on three possible actions is shown in Figure Q-2. If a particular set of actions are never observed consecutively (such as action 1 to action 3 in Figure Q-2), the count (and therefore the probability estimate) for that transition is zero. The size of the state transition matrix is heavily dependent on the number of existing actions (NxN for N actions), and can be large for CBT settings. Even with only three actions in Figure Q-2, it would take a sequence of at least length 10 (9 transitions) to have no zero-probability transitions.

Often the set of actual transitions in any particular training procedure will not cover the entire set of possible transitions. When included in the model, these zero probability events send the KL divergence to infinity. Instead a small (but non-zero) probability can be assigned to transitions that do not occur in the intended procedure. This results in a large divergence score (poor performance) in the PAM but does not send the divergence to infinity. Frequency estimation is a set of statistical techniques that provide estimates

Sequence: 1,1,2,2,3,3

Transition Counts

Starting State \ Ending State	1	2	3
1	1	1	0
2	0	1	1
3	0	0	1

Transition Probabilities

Starting State \ Ending State	1	2	3
1	0.5	0.5	0
2	0	0.5	0.5
3	0	0	1

Figure B-2: Example action sequence with transition counts and probabilities. In the sequence, the numbers represent individual actions performed by the trainee.

of the probabilities of unobserved events, such as the zero-probability events in the transition matrix. Some of the most common frequency estimation methods include additive smoothing and Good-Turing estimation. Additive smoothing simply adds new transition counts such that there are no zero-probability transitions. This strategy works well when there are only a few unobserved events, but can dramatically alter the overall distribution if there are many zero-probability events such as is observed in CBT data. Good-Turing estimation estimates the probability of novel events based on the number of infrequently observed events, and thus self-corrects for cases where the transition matrix is sparse (few non-zero values). Thus for the PAM, Good-Turing smoothing was selected based on its ability to handle large numbers of novel events.

B.1.2 Adherence Metrics as Features

Of the wide range of adherence metrics that can be used as features in machine learning, the Levenshtein distance was selected for its simplicity in interpretation, and the PAM was selected for its ability to meet the four criteria stated earlier in this appendix. PAM is calculated as described above, while Levenshtein distance is calculated by the minimum number of edits (insertions, deletions, or substitutions) to change the trainee sequence into the intended sequence. For each of these metrics, the value after each action was calculated, representing the process-level features. The final value at the end of the module provided the summative-level features.

Adherence metrics such as the PAM or Levenshtein distance can be utilized as features both at a process-level (for each action) or at a summative-level (end of module). While calculating the metric at each action would provide a high number of features for use in machine learning methods, as discussed in Chapter 3, having too many features compared to data points may cause overfitting of the models. Thus, two approaches can be taken: using only summative-level metrics or a reduced set of process-level metrics based on feature selection techniques. In this analysis, both approaches were taken and the results are presented in Chapter 4.

To illustrate the calcuation of these values, consider an example where the prescribed sequence is AABBCC, and the trainee sequence is ABBCDC. In this example, the trainee has made several mistakes: they have omitted an "A" action and have included an extraneous "D" action that does not show up in the prescribed sequence. Both the

Levenshtein distance and PAM can be used as example adherence metrics to describe the timing and severity of the errors. Table Q.1 shows the progression of both the Levenshtein distance and PAM over the course of this sequence. Both Levenshtein and PAM increase as the trainee moves through the sequence and commits the errors. The Levenshtein distance essentially acts to "count" the number of errors made by the trainee. At action 2 where the trainee omits an "A" action and at action 5 where the trainee includes an additional "D" action, the Levenshtein distance increases by one, resulting in a final value of 2. The PAM shows additional fluctuations based on comparing the flow of the sequences as a whole, through the comparison of the transitions between actions. By accounting for transitions (and thus action ordering), there is a relatively small penalty for the PAM at action 4, where the subsequence ABBC is seen in both the trainee and intended sequences. A much harsher penalty is given by the PAM to the added "D" action, as it results in transitions both to and from the "C" action which are unseen in the intended sequence. Both are valid representations of the adherence of the trainee through time, and thus were included as potential features in machine learning approaches. Note that if this sequence comprised the full module, the resultant summative adherence scores would be 2 and 5.99 for Levenshtein distance and PAM, respectively. Both the summative and process-level (action by action) values for the adherence metrics are utilized as possible adherence features in the machine learning sections presented in Chapter 4.

Table B.1: Progression of Levenshtein and PAM over example sequences

Action Number	1	2	3	4	5	6
Intended Action	A	A	B	B	C	C
Trainee Action	A	B	B	C	D	C
Levenshtein Distance	0	1	1	1	2	2
PAM	0	2.04	2.68	1.70	5.06	5.99

Appendix C

Comparison of Sequence Distance Metrics

Because procedures can be seen as a set of actions sequentially ordered towards the accomplishment of a task, they may be described as sequences of steps or events that must occur in a given system ensuring its proper operation. Each sequence contains a number of events, available from an event space, called a vocabulary list. Typically, this vocabulary list is known or can be elaborated through a Cognitive Task Analysis (CTA) [166]. The events generated by a user may not necessarily match in number or type the steps prescribed by the intended protocol, but all possible events are contained in the vocabulary list.

Protocol adherence can be measured by the difference between the procedure expected to be followed and the actual procedure adopted by the user or trainee. If these procedures are represented as sequences, then the task becomes one of determining the difference between two sequences. This section elaborates on existing algorithms used to measure similarities between sequences through two approaches: sequence- and model-based. These represent two commonly implemented types of methods for treating and comparing sequential data. The section closes with the presentation of the proposed algorithm for measuring protocol adherence in training scenarios.

C.1 Sequence-Based Methods

Algorithms that directly employ the sequences of events generated are defined as sequence-based. These are commonly used in string comparison and genetic association. In particular, the Levenshtein distance is introduced as a common sequence-based method from language processing. Another method that uses suffix arrays is also analyzed, being a successful sequence distance metric in online search filters. The usefulness of each of these techniques in measuring procedural adherence is discussed.

C.1.1 Levenshtein Distance

Levenshtein distance is a measure of similarity between two sequences, usually strings of characters. It counts the smallest number of changes required to make the target sequence, equal to the source sequence. It is also known as edit distance. The types of edits allowed are character insertion, deletion and substitution. The Levenshtein algorithm is a valuable tool for genome analysis [167], spell checking and fraud detection [168]. As an illustrative example, consider the following source and target strings: MAED and FADES. The smallest editing distance in this case, according to Levenshtein, is 4. There are three substitutions in the source string ("M", "E" and "D" which are replaced by "F", "D" and "E" respectively) and one character insertion ("S") at the end. Mathematically, the Levenshtein distance between two sequences a and b can be calculated in a recursive function as shown in Equation C.1.

$$lev_{a,b}(i,j) = \begin{cases} max(i,j) & \text{if } min(i,j) = 0 \\ min & \begin{cases} lev_{a,b}(i-1,j) + 1 \\ lev_{a,b}(i,j-1) & otherwise \\ lev_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} \end{cases} \tag{C.1}$$

where $1_{(a_i \neq b_j)}$ is an indicator function equal to zero when $a_i = b_j$ and equal to 1 otherwise.

Modified versions of this algorithm have been widely used, namely with the inclusion of transpositions [168], inversions or both [167]. Transposition is a simple switch of two elements in the same sequence. Inversion is a change in a segment of the sequence that

reverses the order of the elements within the segment. While inversions are unlikely in behavioral data sequences, transpositions may be common occurrences between sequences. The inclusion of transpositions was considered in the Damerau-Levenshtein [168] variant. Using the same example from above, we can recalculate the edit distance according to the Damerau-Levenshteins distance. There is one substitution at the beginning, one transposition (of characters E and D) and one insertion at the end, yielding an edit distance of 3. An overview of the Levenshtein distance and Damerau-Levenshtein measure can be found respectively in [163] and [168].

In case of procedure adherence the Levenshtein measures can quantitatively inform of how many steps were different from the protocol. This is the most common type of objective procedural adherence analysis currently performed in the field. However, it does not account for the nature of any deviations. If there is a step of the procedure that was not supposed to be taken, the Damerau-Levenshtein distance marks this dissimilarity as 1 unit of distance between the two sequences. However, if there is a transposition of two adjacent steps which were both expected to occur, the Damerau-Levenshtein also marks this as a 1 unit of distance. This measurement is then insensitive to the type of deviations and does not penalize event variations differently. In protocol adherence, different steps from the procedure may have markedly different impacts on the safety of a system and thus should be treated differently. Therefore, the standard Levenshtein distance was used in the analyses presented in Chapter 4.

C.1.2 Suffix Arrays

Another algorithm which can be used to identify sequence matches is a method based on suffix arrays. Generally employed in word comparison applications, such as on-line search filters [165], the technique applies mainly to sequences of strings. A suffix array for a set of strings is a lexicographically ordered array for all suffixes of all strings. Consider the following string: "Andromeda". It has 9 suffixes: "andromeda", "ndromeda", "dromeda" and so on, until "da" and "a". Lexicographically, the suffixes are arranged as shown in Table C.1.

Note the alphabetic ordering of the suffixes. If the original string is known, each suffix can be identified by its index. The suffix array is the array of indices of suffixes sorted in lexographical order, which in this case corresponds to: 9,1,8,3,7,6,2,5,4. The

Table C.1: Example of Suffix Array

Index	Sorted Suffix
9	a
1	andromeda
8	da
3	dromeda
7	eda
6	meda
2	ndromeda
5	omeda
4	romedae

lexicographic ordering allows similar substrings to be grouped together for comparison. Malde et al. [169] decomposed the Suffix Array algorithm into three stages. The first one identifies pairs of sequences that contain matching blocks. Two sequences have a matching block if they have contiguous segments that match exactly. The second step uses the information generated in the previous process to calculate a score for pairs of sequences, which in the final stage, will be used to perform hierarchical clustering.

The score of a pair of sequences defines a measure of their similarity. Consistent matching blocks are a set of non-overlapping and co-linear (i.e. arranged in the same order) matching blocks in the sequences. Because this section focuses on establishing a metric for adherence, only the first and second stage of this method will be addressed. The scoring process comprises of the following substeps:

```
Create Suffix;
Sort Suffix into SuffixArray;
Clique = Group of Suffixes that share a prefix;
for (Each Clique) {
    Find maximal matching blocks
    if (#MatchingBlocks >= 1) {
        Group matching blocks
    }
}
Score = Sum of largest set of matching blocks
```

A description of the Suffix Array method is given by Manber & Myers [165] and Malde [169] explains its usefulness in clustering sequences. Clustering is performed hi-

erarchically, with rules for cluster merging and creation of new leaves. This results in a set of clusters that are binary trees, with the sequences as leaves. The lowest branches represent the strongest connected sequences, decreasing in strength of connection further up the tree. For example, consider the sequences shown in Table C.2.

Using the suffix array method, we can find many matching blocks between the target sequence and Source 1. The largest matching blocks are of length 4 (AABB, BBCC, CCDD). However, the longest set of consistent blocks is AABB and CCDD. Therefore, the suffix array method would assign a score of 8 to this sequence.

The Suffix Array approach only accounts for the consistent matching blocks when generating scores for pairs of sequences (non-matching blocks are ignored). This represents a considerable weakness of this method in human behavior applications. In the example above, it is apparent that Source 1 did not exactly follow the target sequence. However, the suffix-array method assigns a score of 8 to this source sequence, which is the same score as if the user had performed the intended sequence exactly. It is important for procedural adherence purposes for the metric to be able to distinguish between perfect performance and sequences such as Source 1 given above.

An even more salient example arises if one considers the Source 2 sequence from above. In this sequence, the middle section has been substituted for four "E states that do not exist in the target sequence. However, the same AABB and CCDD subsequences are still intact, and the suffix array method would still assign a score of 8 to this source sequence. For human behavioral analysis and procedural adherence measurement, it would be very important to be able to distinguish a user that followed the target sequence exactly from the two example source sequences described above. In short, the suffix array method identifies the similarities between two sequences, but ignores the number and nature of deviations. This makes it less desirable as a metric for measuring procedural adherence in training.

In summary, sequence-based algorithms track mainly the number of similarities or

Table C.2: Example Sequences for Suffix Array Comparison

Type	**Sequence**
Target	AABBCCDD
Source 1	AAAABBBBCCCCDDDD
Source 2	AAAABBEEEECCDDDD

deviations between two sequences but suffer from a major shortcoming in that they are insensitive to the nature of deviations. In contrast, an alternate set of methods use models to account for the nature of deviations between sequences and are described in the next section.

C.2 Model-Based Methods

In behavioral assessment it is important not only to account for the number of protocol deviations, but also how the events deviated from the expected. To accomplish this, sequences can be modeled as a set of transition probabilities between events. In a training scenario, these events are known and observable to the system user.

Model-based methods offer particular advantages in training scenarios: they are sensitive in changes in the sequence of states (based on changes in the transitions between states) and can predict the next event based on the current one. Mathematically this can be expressed by Markov models, representing the series of events as a succession of states.

The simplest instantiation of Markov models are discrete Markov chains [170], which can be used to represent the set of operator events as N different states $S_1, S_2, , S_N$. The states are linked by a set of transition probabilities $A = a_ij$. Between time t and $t+1$, the system undergoes a state transition, possibly returning back to the same state, according to these probabilities. An illustration of an example system is given in Figure D-1. Being a discrete process, the instants at which the states change are designated as $t = 1, 2, \ldots$ and the present state at time t as s_t. The model in question follows first-order Markov assumption, which states that the current state s_t at time t is only dependent on the value of the system at time $t - 1$ and no other value before. Formally, this probability description at time t can be expressed as shown in Equation C.2.

$$P(s_t = S_j | s_{t-1} = S_i, s_{t-2} = S_k, \ldots) = P(s_t = S_j | s_{t-1} = S_i) = a_{ij}(t), 1i, jN \qquad \text{(C.2)}$$

Assuming that the transition probabilities a_{ij} are stationary (time-independent), the Markov assumption holds so that:

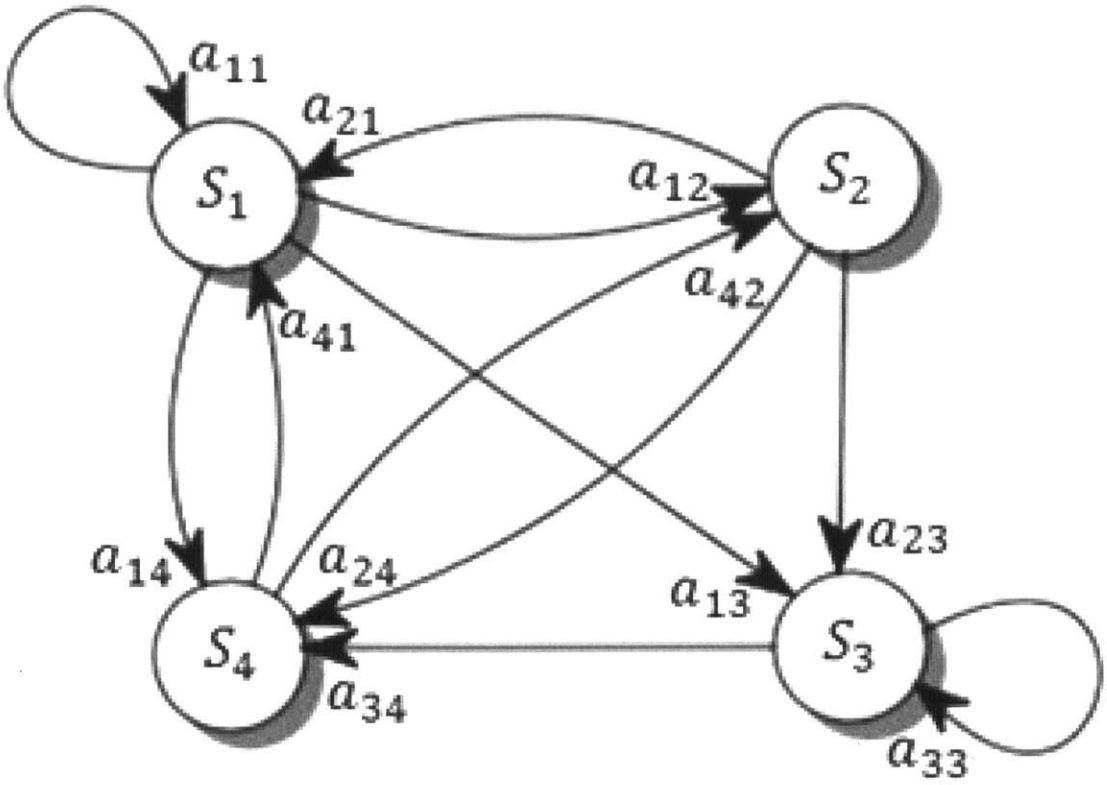


Figure C-1: Markov chain with four states and the respective transitions probabilities

$$P(s_t = S_j s_{t-1} = S_i) = a_{ij}(t) \geq 0, 1 \leq i, j \leq N and \sum_{j=1}^{N} a_{ij} = 1 \tag{C.3}$$

An overview of Markov Chains can be found in [170]. In procedural training, it is assumed that the states are observable, and a Markov Chain can be constructed for each sequence using transition probabilities learned from the sequence.

In contrast with sequence-based algorithms which use sequences directly, there are algorithms capable of measuring the distance between two models from a statistical standpoint. Such methodologies can then be used to approximate the statistical distance between two sequences as the distance between two models trained on those sequences.

C.2.1 Kullback-Leibler Divergence

The Kullback-Leibler (KL) divergence, also known as relative entropy, attempts to provide a quantification of statistical distance between models or probability distributions

[164]. It has been previously applied to Hidden Markov Models (HMMs), which are a more complex type of generative Markov models. In Hidden Markov Models, the states are hidden, the user has only access to an output set of observable variables [170–172]. In proceduralized settings, the states are assumed to be observable, and thus are more appropriately fit by a Markov Chain. While the KL divergence has not previously been applied to Markov Chains, a similar strategy to HMMs can be used.

To apply the KL divergence to Markov Chains, a set of models Λ are first trained for each of the data sequences T. A likelihood matrix L is obtained from modeled sequences and whose element l_ij defined as shown in Equation C.4.

$$l_ij = \frac{1}{length(T_j)} log_2 f_T(T_j; \lambda_i), 1 \leq i, j \leq N \tag{C.4}$$

where T_j is the jth sequence, λ_i is the ith model trained for the ith sequence and $f_T(\cdot\,; \lambda_i)$ is the probability density function (pdf) over sequences according to the model λ_i. This is to say that the ith column of l represents the likelihood of sequence T_j under each of the trained models.

By normalizing L such that each column adds up to one, the columns become probability density functions over the trained model space Λ of each individual sequence. The formulation for the discrete case is given in Equation .

$$D_{KL}(f_P || f_Q) = \sum_i f_P(i) log \frac{f_P(i)}{f_Q(i)} \tag{C.5}$$

where f_P and f_Q are two discrete pdfs representing Markov models based on the sequences P and Q. Note that due to the normalization, the divergence becomes now the measure of dissimilarity between pdfs. This KL divergence is also asymmetric, so by symmetrizing it, one obtains Equation C.6.

$$D_{KL_SYM}(f_P || f_Q) = \frac{1}{2}[D_{KL}(f_P || f_Q) + D_KL(f_Q || f_P)] \tag{C.6}$$

In short, the KL distance is based on the calculation of the likelihood of the sequences given the models. A similar model-based algorithm is the BP metric [171, 172]. The difference between these two metrics lies in the fact that the BP bases its sequence distance on the likelihood matrix L, instead of normalizing and treating the likelihood data as probability density functions. Both consider the quality of modeling fit of each

trained model to a sequence by estimating the normalized difference between the sequence and the training likelihood [171, 172]. However, the KL metric was selected based on having shown superior clustering performance in the literature [164]. Table C.3 presents a comparison of the presented metrics with respect to particular qualities of interest for measuring sequence distance. As shown in this table, the KL divergence meets all of the needs for a procedure adherence metric, and therefore was used as the basis for the PAM.

Table C.3: Comparison of Sequence Distance Metrics

Characteristics	**Levenshtein Distance**	**Suffix Array**	**Kullback-Leibler**
Variable importance of errors	No	No	Yes
Non-equal sequence length	Yes	Yes	Yes
Sensitivity to number of deviations	Yes	No	Yes
Sensitivity to transitions between events	No	No	Yes

Appendix D

Selection of KL Divergence Features

A set of potential features were identified for the calculation of the PAM metric based on the KL divergence over the course of a training module. These included the terminal (final) KL value at the end of the module, the mean KL value, the median KL value, the area under the KL graph over the course of the module, the peak (maximum) KL value, and the ratio of the terminal to peak value. In order to identify the features that could be the most informative of good/poor trainees, the subjects were ranked from best to worst performers based on each feature. In comparison, an expert with full knowledge of the procedure was asked to analyze the volunteers data to act as the gold standard. The expert ranked the volunteers from top to worst performers based on subjective assessment of their performance. The sum of rank differences between the feature rankings and the expert rankings were calculated for each feature. The area feature was found to have the lowest sum of differences, and this was statistically significant from all other features other than mean by Mann-Whitney U Test. Figure ?? shows the sum of rank differences for each feature, and Table D.1 shows the statistical test results. Based on this analysis, the area below the graph was the selected feature of the KL to be used in the score computation.

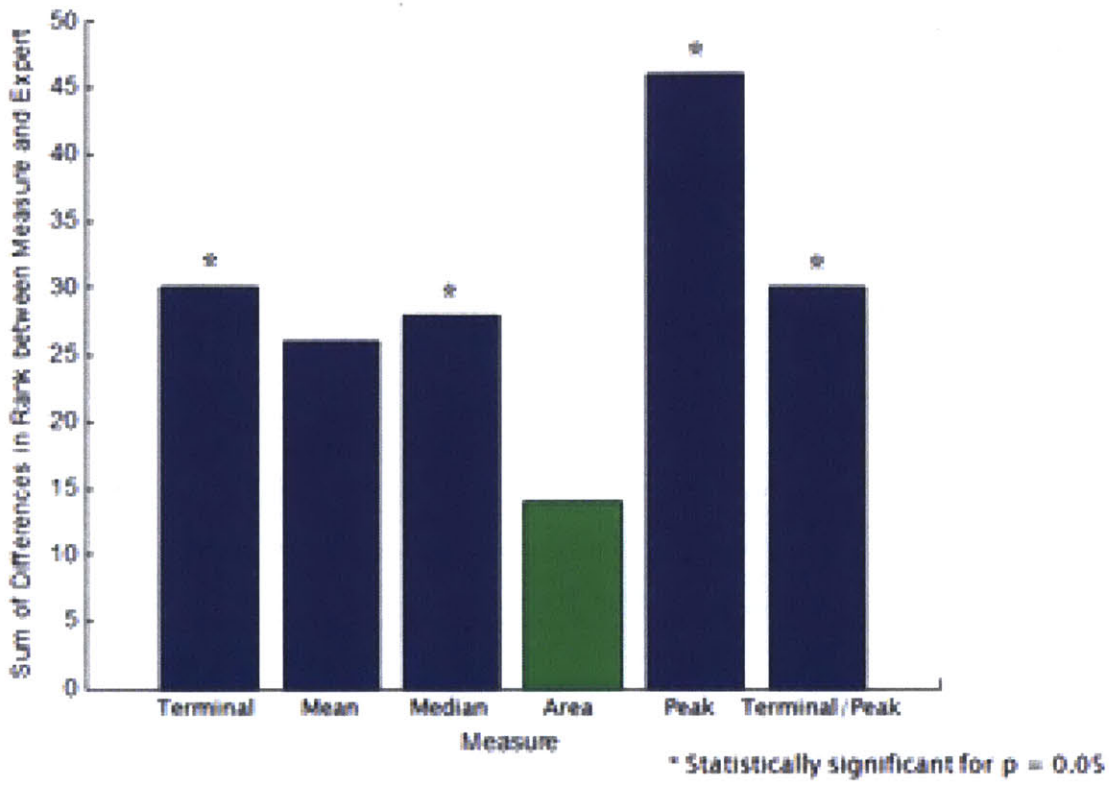


Figure D-1: Graph of the sum of rank differences between each feature and the expert ranking. P values indicate significant difference from the lowest error (Area) by Mann-Whitney U test.

Table D.1: Statistical analysis of varying features of the KL graph. The values were calculated by comparing the sum of rank differences of each feature to the sum of rank differences based on area.

Feature	**P-value of Mann-Whitney U Test**
Terminal Value	0.0394
Area	N/A
Mean	0.0541
Median	0.0322
Peak	0.0042
Terminal/Peak	0.0121

Appendix E

Rule-Based Experiment Procedures

This appendix provides the procedures used in the simluated nuclear reactor Computer-Based Training (CBT) experiment. The procedures for each training and test module contained different content, but maintained a similar structure. There were 4 types of procedures contained in the experiment, as noted by the procedure number in the upper left corner of the procedure. These were:

- E = Emergency procedures, used in the case of a reactor SCRAM
- M = Maintenance procedures, used for contacting and coordinating with maintenance
- AOP = Anomalous Operating Procedures, used in case of unusual behavior but no SCRAM
- T = Training procedures, used for training modules

For the training modules, the participants were only provided the associated training procedures. In the test module, the participants were provided a binder containing the E, M, and AOP procedures, as well as appendices providing department codes nominal value ranges for the system parameters. In the following pages, the training procedures are presented first, followed by the binder used in the test module.

A. PURPOSE

This training procedure provides actions to shut down the active reactor, to start up the inactive reactor, and to verify the conditions in which the reactor is being shut down or started up.

B. SYMPTOMS OR ENTRY CONDITIONS

Condition	Description
1	Check Rod Insertion State – 0%
2	Check ALL Power Output > 0

Step	Action/Expected Response	Response Not Obtained
1	**Initiate Rod Insertion**	
	-Set rod insertion control – AUTO ON	
	-Set rod insertion rate – 50 steps/min	
	-Initiate rod insertion	
	-Verify rod insertion	-Manually insert rods – 25 steps/min
2	**Verify Power Reduction**	
	-Reactor Temp – DROPPING	-Go to Step 1
	-RCS Pressure – DROPPING	
	-RCS Temp – DROPPING	
	-ALL SG Pressure – DROPPING	
	-ALL SG Temperature - DROPPING	
	-ALL SG Coolant Level – STABLE OR DROPPING	
	-ALL Power Output - DROPPING	
3	**Halt Power Production**	
	-ALL RCPs to Intact SGs – DEACTIVATED	-Deactivate RCPs to intact SGs
	-Verify Power Output – DROPPING	
	-Verify Turbine Steam Press – DROPPING	
	-Verify SG Temperature – DROPPING	
	-Verify SG Pressure – DROPPING	
	-Intact SGs Steamline Valves – CLOSED	-Close SGs Steamline Valves
	-Intact SGs Secondary Coolant Pumps DEACTIVATED	-Deactivate SGs Secondary Coolant Pumps
	-Deactivate Turbines for intact loops – indicators LIT RED	
	-Verify Expected Alarm – B3 Turbine Auto Trip LIT RED	
4	**Initiate Rod Extraction**	
	-Verify rod extraction control – AUTO ON	-Set rod insertion control AUTO ON
	-Verify rod extraction rate – 50 steps/min	-Set rod insertion rate – 50 steps/min
	-Initiate rod extraction	
	-Verify rod extraction	-Manually extract rods – 25 steps/min

Step	Action/Expected Response	Response Not Obtained
5	**Initiate Power Production**	
	-Intact SGs RCPs – ACTIVATED	-Activate intact SGs RCPs
	-Intact SGs Steamline Valves – OPEN	-Open intact SGs Steamline Valves
	-Intact SGs Secondary Coolant Pumps ACTIVATED	-Activate intact SGs Secondary Coolant Pumps
	-Activate Turbines for intact loops – indicators LIT GREEN	

Number	Title	Rev./Date
T-2	**Managing Coolant Flow**	Rev.9 8/15/2011

PURPOSE

A. This training procedure provides actions to manage the coolant system of the reactor, which involves both the reactor coolant pumps, secondary coolant pumps, and the various coolant flows and levels associated with them. The procedure provides several goals that may be achieved by managing the coolant system, given as examples of tasks that may be required of a reactor operator.

Goals are accomplished by the steps in each box. Do not go on to the next step or box until the goal condition has been met.

B. SYMPTOMS OR ENTRY CONDITIONS

Condition	Description
1	Check Rod Insertion State – 0%
2	Check ALL Power Output > 0
3	Check ALL RCPs – ACTIVATED
4	Check SI Trains - ACTIVATED
5	Check ALL SCPs – ACTIVATED

Number	Title	Rev./Date
T-2	**Managing Coolant Flow**	Rev.9 8/15/2011

Step	Action/Expected Response	Response Not Obtained

GOAL: Lower Loop 1 SG Temperature to under 450 degrees

1 **Deactivate Loop 1 RCP**
- -Deactivate Loop 1 RCP
- -Verify SG Pressure – DROPPING
- -Verify SG Temp – DROPPING
- -Verify Turbine Steam Pressure – DROPPING
- -Verify Power Output –DROPPING

GOAL: Raise Reactor Coolant Level to at least 75.

2 **Activate SI Trains**
- -Activate ALL SI Train pumps
- -Verify Reactor Temp – STABLE OR DROPPING
- -Verify RCS Temp – STABLE OR DROPPING
- -Verify RCS Pressure – STABLE OR DROPPING
- -Verify Coolant Level – STABLE OR RISING
- -Verify Expected Alarm – A3 SI Auto Injection LIT RED

3 **Deactivate SI Trains**
- -Deactivate SI Train A pump
- -Deactivate SI Train B pump

4 **Activate Loop 1 RCP**
- -Activate Loop 1 RCP

GOAL: Raise Loop 2 Turbine Steam Pressure to at least 865.

5 **Deactivate Loop 2 Cooling**
- -Deactivate Loop 2 Secondary Coolant Pump
- -Verify Coolant Temp – STABLE or RISING
- -Verify Turbine Steam Pressure – STABLE or RISING
- -Verify SG Temperature – STABLE OR RISING
- -Verify SG Coolant Level – STABLE OR DROPPING

6 **Activate Loop 2 Cooling**
- -Activate Loop 2 Secondary Coolant Pump

GOAL: Lower Loop 3 SG Pressure to less than 1000.

7 **Open Loop 3 Steam Dump Valve**
- -Open Loop 3 Steam Dump Valve
- -Verify SG Pressure – DROPPPING
- -Verify SG Temperature – DROPPING
- -Verify Turbine Steam Pressure – DROPPING
- -Verify Power Output – DROPPING

8 **Close Steam Dump Valves**
- -Loop 3 Steam Dump Valve – CLOSED

A. PURPOSE

This training procedure provides actions to diagnose the cause of reactor or loop system malfunctions following an automatic scram of the reactor.

B. SYMPTOMS OR ENTRY CONDITIONS

Condition	Description
1	Check Rod Insertion State – 100%
2	Check Reactor Scram Annunciator (A2) – LIT

Step	Action/Expected Response	Response Not Obtained
1	**Verify Reactor Scram** -Verify all rods at 100% insertion -Verify neutron flux – DROPPING	-DO NOT CONTINUE MODULE
2	**Verify Turbine Trip** -All turbine RED indicators - LIT	-Manually switch indicators to RED
3	**Verify SI Status** -Verify SI system is actuated: Annunciator A3 LIT -Verify both trains of SI – ACTUATED -Verify both SI Pumps – ACTIVATED	-Check if SI is required: If RCS pressure is less than 1807 psig
4	**Verify Containment Pressure HAS REMAINED LESS THAN 30 PSIG**	-Stop all RCPs -Place steam dumps in steam pressure mode -Verify CONTAINMENT ISOLATION WARNING: Annunciator A4 - LIT
5	**Verify Secondary Coolant Flow for ALL Loops – GREATER THAN 720 GPM**	-Ensure secondary coolant pumps are activated -Stop dumping steam
6	**Verify RCS Temperature** -IF any RCP running, THEN verify RCS average temperature – STABLE BETWEEN 557° AND 562° -IF no RCP running, THEN verify reactor coolant temperature - STABLE BETWEEN 557° AND 562°	-IF temperature less than 557° and dropping, THEN: a) Close any open steam dump valves b) IF cooldown continues, THEN close main steamline valves -IF temperature greater than 562° and rising, THEN open ALL steam dump valves

Step	Action/Expected Response	Response Not Obtained
7	**Check if SGs are not faulted** -Verify all SGs – PRESSURIZED	-IDENTIFY FAULT AS FAULTED STEAM GENERATOR ON POST-TEST -THIS ENDS THE MODULE- DO NOT CONTINUE
8	**Check if SG Tubes are not ruptured** -Verify condenser radiation level – NORMAL	-IDENTIFY FAULT AS STEAM GENERATOR TUBE RUPTURE ON POST-TEST -THIS ENDS THE MODULE- DO NOT CONTINUE
9	**Check if RCS is intact** -Verify containment radiation – NORMAL -Verify containment pressure – NORMAL -Verify containment temperature – NORMAL	-IDENTIFY FAULT AS LOSS OF REACTOR OR SECONDARY COOLANT ON POST-TEST -THIS ENDS THE MODULE- DO NOT CONTINUE
10	**Check if SI Trains Reset** -Verify Train A sequencer status lights – LIT GREEN -Verify Train B sequencer status lights – LIT GREEN	-Go to Step 1

NUCLEAR POWER PLANT OPERATING MANUAL

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Nuclear Reactor Operating Procedures

Maintenance Procedures

M-1	**Contacting Departments for Maintenance**

Emergency Procedures

E-0	**Reactor Scram or Safety Injection**
E-1	**Loss of Reactor or Secondary Coolant**
E-2	**Faulted Steam Generator Isolation**
E-3	**Steam Generator Tube Rupture**

Anomalous Operating Procedures

AOP-1	**Power Excursion**
AOP-2	**Loop Data Loss**
AOP-3	**Reactor Data Loss**
AOP-4	**Rod Insertion Data Loss**
AOP-5	**Turbine Power Generation Malfunction**
AOP-6	**Loop Temperature/Pressure Anomaly**

Appendices

Appendix A	**Normal Operating Meter Values**
Appendix B	**Power Plant Departments**

Number	Title	Rev./Date
M-1	Contacting Departments for Maintenance	Rev.6 6/17/2011

A. PURPOSE

This Maintenance Procedure provides actions contact different departments for maintenance purposes.

B. SYMPTOMS OR ENTRY CONDITIONS

Condition	Description
1	Operator is directed by a procedure to contact a department.

Number	Title	Rev./Date
M-1	Contacting Departments for Maintenance	Rev.6 6/17/2011

Step	Action/Expected Response	Response Not Obtained
1	Type the NAME OF THE DEPARTMENT in the chatbox and press enter.	Wait for response
2	Type the ERROR CODE in the chatbox and press enter.	
3	Wait for confirmation.	

Number	Title	Rev./Date
E-0	Reactor Scram or Safety Injection	Rev.1

Purpose

This procedure provides actions to verify proper response of the automatic protection systems following manual or automatic actuation of a reactor scram or safety injection, to assess plant conditions, and to identify the appropriate recovery procedure.

Symptoms or Entry Conditions

The following are symptoms of a reactor scram and/or safety injection:

1. Reactor scram annunciator (A2) lit.
2. All control rods fully inserted. Rod bottom lights lit.
3. Annunciator A3 lit.
4. SI Train A and/or B pumps activated.

FOLDOUT PAGE FOR E-0

1. RCP TRIP CRITERIA

   a. Trip ALL Reactor Coolant Pumps (RCPs) if BOTH conditions below exist:

      a. AT LEAST ONE Safety Injection (SI) pump is running.

      b. RCS pressure LESS THAN 1360 PSIG.

2. CONTINUOUS ACTION STEPS

   Check Containment Pressure (Step 4)

   Check RCS Temperature (Step 6)

   Check if RCPs should be stopped (Step 7)

STEP	ACTION / EXPECTED RESPONSE	RESPONSE NOT OBTAINED
1	**Verify Reactor Scram:** • Rod bottom lights – LIT • Reactor scram annunciator (A2) – LIT • Neutron flux - DROPPING	
2	**Verify Turbine Trip:** a. All turbine RED indicators - LIT	a. Manually switch indicators to RED
3	**Check SI Status:** a. Check if SI is actuated: Annunciator A3 – LIT	a. Check if SI is required: If RCS pressure is less than 1807 psig
	b. Verify both trains of SI – ACTUATED Train A & B sequencer status lights – LIT RED Both SI Pumps activated	b. Manually actuate SI
4	**Check Containment Pressure HAS REMAINED LESS THAN 30 PSIG**	Perform the following: a. Stop all RCPs b. Place steam dumps in steam pressure mode c. Verify CONTAINMENT ISOLATION WARNING: Annunciator A4 – LIT

FOLDOUT PAGE FOR E-0

1. RCP TRIP CRITERIA

   a. Trip ALL Reactor Coolant Pumps (RCPs) if BOTH conditions below exist:

      a. AT LEAST ONE Safety Injection (SI) pump is running.

      b. RCS pressure LESS THAN 1360 PSIG.

2. CONTINUOUS ACTION STEPS

   Check Containment Pressure (Step 4)

   Check RCS Temperature (Step 6)

   Check if RCPs should be stopped (Step 7)

STEP	ACTION / EXPECTED RESPONSE	RESPONSE NOT OBTAINED
5	**Verify Secondary Coolant Flow for ALL Loops – GREATER THAN 720 GPM**	IF flow less than 720 GPM, THEN: a. Ensure secondary coolant pumps are activated b. Stop dumping steam
6	**Check RCS Temperature:** • IF any RCP running, THEN check RCS average temperature – STABLE BETWEEN 557° AND 562° OR • IF no RCP running, THEN check reactor coolant temperature – STABLE BETWEEN 557° AND 562°	IF temperature less than 557° and dropping, THEN: a. Stop dumping steam b. IF cooldown continues, THEN close main steamline valves IF temperature greater than 562° and rising, THEN open ALL steam dump valves
7	**Check if RCPs should be stopped:** a. Check RCPs – ANY RUNNING b. RCS pressure – LESS THAN 1360 psig c. Stop all RCPs d. Place steam dumps in steam pressure mode	a. Verify steam dumps in steam pressure mode. Go to Step 8 b. Go to Step 8 c.

FOLDOUT PAGE FOR E-0

1. RCP TRIP CRITERIA

   a. Trip ALL Reactor Coolant Pumps (RCPs) if BOTH conditions below exist:

      a. AT LEAST ONE Safety Injection (SI) pump is running.

      b. RCS pressure LESS THAN 1360 PSIG.

2. CONTINUOUS ACTION STEPS

   Check Containment Pressure (Step 4)

   Check RCS Temperature (Step 6)

   Check if RCPs should be stopped (Step 7)

8	**Check if SGs are not faulted:** All SGs - PRESSURIZED	IF pressure in any SG is dropping in an uncontrolled manner OR any SG is depressurized, THEN go to **E-2, FAULTED STEAM GENERATOR ISOLATION, Step 1**

FOLDOUT PAGE FOR E-0

1. RCP TRIP CRITERIA

    1. Trip ALL Reactor Coolant Pumps (RCPs) if BOTH conditions below exist:

        a. AT LEAST ONE Safety Injection (SI) pump is running.

        b. RCS pressure LESS THAN 1360 PSIG.

2. CONTINUOUS ACTION STEPS

    Check Containment Pressure (Step 4)

    Check RCS Temperature (Step 6)

    Check if RCPs should be stopped (Step 7)

STEP	ACTION / EXPECTED RESPONSE	RESPONSE NOT OBTAINED
9	**Check if SG Tubes are not ruptured:** a. Condenser radiation level – NORMAL	Go to **E-3, STEAM GENERATOR TUBE RUPTURE, Step 1**
10	**Check if RCS is intact:** • Containment radiation – NORMAL • Containment pressure – NORMAL • Containment temperature - NORMAL	Go to **E-1, LOSS OF REACTOR OR SECONDARY COOLANT, Step 1**
11	**Check if SI Trains Reset:** a. Verify both trains of SI – RESET Train A sequencer status lights – LIT GREEN Train B sequencer status lights – LIT GREEN	Go to Step 1

Number	Title	Rev./Date
E-1	Loss of Reactor or Secondary Coolant	Rev.1

Purpose

This procedure provides actions to terminate leakage or loss of reactor coolant or secondary coolant following a reactor scram and an abnormal containment reading.

Symptoms or Entry Conditions

This procedure is entered from:

1. E-0, REACTOR SCRAM OR SAFETY INJECTION, Step 10, when containment radiation, containment pressure, or containment temperature is abnormal.

FOLDOUT PAGE FOR E-1

1. SI REINITIATION CRITERIA

Manually stop SI pumps if RCS core exit temperature – LESS THAN 25° F

2. SECONDARY INTEGRITY CRITERIA

Go to E-2, FAULTED STEAM GENERATOR ISOLATION, Step 1, if any SG pressure is dropping in an uncontrolled manner or if any SG has completely depressurized, and that SG has not been isolated.

3. CONTINUOUS ACTION STEPS

Check if RCPs should be stopped (Step 1)

Check Intact SG(s) Levels (Step 3)

Check Secondary Radiation – NORMAL (Step 4)

STEP	ACTION / EXPECTED RESPONSE	RESPONSE NOT OBTAINED
1	**Check if RCPs should be stopped:**	
	a. Check RCPs – ANY RUNNING	a. Verify steam dumps in steam pressure mode. Go to Step 2
	b. RCS pressure – LESS THAN 1360 psig	b. Go to Step 2
	c. Stop all RCPs	
	d. Place steam dumps in steam pressure mode	
2	**Check if SGs are not faulted:**	IF pressure in any SG is dropping in an uncontrolled manner OR any SG is depressurized, THEN go to **E-2, FAULTED STEAM GENERATOR ISOLATION, Step 1**
	a. All SGs - PRESSURIZED	
3	**Check Intact SG Levels:**	
	a. Coolant Level – GREATER THAN 7%	a. IF coolant level LESS THAN 7%, THEN engage secondary coolant pump until SG level greater than 7%
	b. Control secondary coolant pump to maintain level between 7% and 70%	b. IF level in any intact SG continues to rise in an uncontrolled manner, THEN go to **E-3, STEAM GENERATOR TUBE RUPTURE, Step 1**
4	**Check Secondary Radiation – NORMAL**	Go to **E-3, STEAM GENERATOR TUBE RUPTURE, Step 1**
	a. Condenser radiation – NORMAL	

FOLDOUT PAGE FOR E-1

1. SI REINITIATION CRITERIA

   Manually stop SI pumps if RCS core exit temperature – LESS THAN 25° F

2. SECONDARY INTEGRITY CRITERIA

   Go to E-2, FAULTED STEAM GENERATOR ISOLATION, Step 1, if any SG pressure is dropping in an uncontrolled manner or if any SG has completely depressurized, and that SG has not been isolated.

3. CONTINUOUS ACTION STEPS

   Check if RCPs should be stopped (Step 1)

   Check Intact SG(s) Levels (Step 3)

   Check Secondary Radiation – NORMAL (Step 4)

STEP	ACTION / EXPECTED RESPONSE	RESPONSE NOT OBTAINED
5	**Check if SI Flow Should be Terminated:**	
	a. RCS Pressure – STABLE OR RISING	a. Go to Step 6
	b. At least ONE intact SG coolant level – GREATER THAN 7%	b. Go to Step 6
6	**Check RCS and SG Pressures:** • Check pressure in all SGs – STABLE OR RISING • Check RCS pressure – STABLE OR DROPPING	Go to **E-0, REACTOR SCRAM OR SAFETY INJECTION, Step 1**
7	**Check if RCS Cooldown and Depressurization is Required:** • RCS Pressure – GREATER THAN 250 PSIG • RCS Temperature – GREATER THAN 390°	Go to Step 8
8	**Check if SI Trains Should be Reset:**	
	a. At least two SG temperatures – LESS THAN 390°	a. Continue with Step 9. WHEN at least two SG temperatures less than 390°, THEN perform Step 8b.
	b. Reset SI Trains – STATUS LIGHT GREEN	
9	**Check if Intact SG(s) Should be Depressurized to RCS Pressure:**	
	a. RCS pressure – LESS THAN INTACT SG PRESSURES	a. Go to Step 10

Step 9 continued on next page

FOLDOUT PAGE FOR E-1

1. SI REINITIATION CRITERIA

   Manually stop SI pumps if RCS core exit temperature – LESS THAN 25° F

2. SECONDARY INTEGRITY CRITERIA

   Go to E-2, FAULTED STEAM GENERATOR ISOLATION, Step 1, if any SG pressure is dropping in an uncontrolled manner or if any SG has completely depressurized, and that SG has not been isolated.

3. CONTINUOUS ACTION STEPS

   Check if RCPs should be stopped (Step 1)

   Check Intact SG(s) Levels (Step 3)

   Check Secondary Radiation – NORMAL (Step 4)

STEP	ACTION / EXPECTED RESPONSE	RESPONSE NOT OBTAINED
	Step 9 continued	
	b. Check SG radiation levels • Condenser radiation levels – NORMAL	b. Do not dump steam from an SG with an unacceptable radiation reading.
	c. Dump steam to condenser from intact SG(s) until SG pressure less than RCS pressure	
10	**Check if SI Flow Should be Terminated:** a. RCS Pressure – STABLE OR RISING	a. Continue SI
	b. At least ONE intact SG coolant level – GREATER THAN 7%	
11	**Stop Both SI Pumps**	
12	**Check RCP Status:**	
	a. Verify RCP(s) to intact SG(s) are running	a. Manually start RCP(s) for intact SG(s)
	b. Verify RCS Pressure and Temperature – NORMAL	
13	**Initiate Rod Extraction:**	
	a. Place rod extraction control in AUTO	
	b. Set rod extraction rate – 10 STEPS PER MINUTE	

Step 13 continued on next page

FOLDOUT PAGE FOR E-1

1. SI REINITIATION CRITERIA

   Manually stop SI pumps if RCS core exit temperature – LESS THAN 25° F

2. SECONDARY INTEGRITY CRITERIA

   Go to E-2, FAULTED STEAM GENERATOR ISOLATION, Step 1, if any SG pressure is dropping in an uncontrolled manner or if any SG has completely depressurized, and that SG has not been isolated.

3. CONTINUOUS ACTION STEPS

   Check if RCPs should be stopped (Step 1)

   Check Intact SG(s) Levels (Step 3)

   Check Secondary Radiation – NORMAL (Step 4)

STEP	ACTION / EXPECTED RESPONSE	RESPONSE NOT OBTAINED
	Step 13 continued	
	c. Verify rod extraction	c. Manually extract rods – 10 STEPS PER MINUTE
14	**Initiate Power Production:**	
	a. Verify intact SG(s) Temperature – NORMAL	a. Verify intact SG(s) RCPs – ACTIVATED
	b. Intact SG(s) Steamline Valves – OPEN	b. Open intact SG(s) Steamline Valves
	c. Intact SG(s) Secondary Coolant Pumps – ACTIVATED	c. Activate intact SG(s) Secondary Coolant Pumps
	d. Place Steam Dump Valves in CLOSED position	
	e. ACTIVATE turbine(s) for intact loop(s) – Indicators LIT GREEN	

- END –

Number	Title	Rev./Date
E-2	**Faulted Steam Generator Isolation**	Rev.1

Purpose

This procedure provides actions to identify and isolate a faulted steam generator following a reactor scram and uncontrolled or full depressurization of one or more steam generators.

Symptoms or Entry Conditions

This procedure is entered from:

1. E-0, REACTOR SCRAM OR SAFETY INJECTION, Step 8, when one or more steam generators are depressurized or depressurizing.
2. E-1, LOSS OF REACTOR OR SECONDARY COOLANT, Step 2, when any SG is dropping in an uncontrolled manner OR any SG is depressurized.

FOLDOUT PAGE FOR E-2

1. SI REINITIATION CRITERIA

   Manually stop SI pumps if RCS core exit temperature – LESS THAN 25°

2. MULTIPLE FAULTED STEAM GENERATOR CRITERIA

   Stabilize the plant by returning to E-2, FAULTED STEAM GENERATOR, Step 1, if any intact SG level falls in an uncontrolled manner or any intact SG has abnormal radiation.

3. CONTINUOUS ACTION STEPS

   Identify Faulted SG(s) (Step 2)

   Check Secondary Radiation (Step 7)

   Check if SI should be terminated (Step 10)

STEP	ACTION / EXPECTED RESPONSE	RESPONSE NOT OBTAINED
1	**Check if Any SG Not Faulted:**	
	a. Check pressures in all SGs – WITHIN LIMITS	a. IF all SG pressures dropping in an uncontrolled manner, THEN go to **E-0, REACTOR SCRAM OR SAFETY INJECTION, Step 1**
2	**Identify Faulted SG(s):**	
	a. Check pressures in all SGs: • ANY SG PRESSURE DROPPING IN AN UNCONTROLLED MANNER OR • ANY SG COMPLETELY DEPRESSURIZED	a. WHEN faulted SG(s) are identified, THEN perform Step 3.
3	**Isolate Faulted SG(s):**	
	a. Place steam dump valve(s) from ruptured SG(s) in CLOSED position	
	b. Check steamline valve(s) from ruptured SG(s) – CLOSED	b. Manually close ruptured SG steamline valve(s)
	c. Check secondary coolant pump(s) from ruptured SG(s) – STOPPED	c. Manually stop ruptured SG secondary coolant pump(s)
	d. Verify RCP from ruptured SG(s) – STOPPED	d. Manually stop ruptured SG RCP(s)

FOLDOUT PAGE FOR E-2

1. SI REINITIATION CRITERIA

Manually stop SI pumps if RCS core exit temperature – LESS THAN 25°

2. MULTIPLE FAULTED STEAM GENERATOR CRITERIA

Stabilize the plant by returning to E-2, FAULTED STEAM GENERATOR, Step 1, if any intact SG level falls in an uncontrolled manner or any intact SG has abnormal radiation.

3. CONTINUOUS ACTION STEPS

Identify Faulted SG(s) (Step 2)

Check Secondary Radiation (Step 7)

Check if SI should be terminated (Step 10)

STEP	ACTION / EXPECTED RESPONSE	RESPONSE NOT OBTAINED
4	**Check Remaining SG Levels – GREATER THAN 9%**	Go to **E-0, REACTOR SCRAM OR SAFETY INJECTION, Step 1**
5	**Reset SI Trains A and B**	
6	**Check Containment Levels:**	
	a. Containment Radiation – NORMAL	a. Manually close ruptured SG steamline valve(s)
	b. Containment Pressure – NORMAL	b. Manually stop ruptured SG secondary coolant pump(s)
	c. Containment Temperature - NORMAL	c. Manually stop ruptured SG RCP(s)
7	**Check Secondary Radiation:**	Go to **E-3, STEAM GENERATOR TUBE RUPTURE, Step 1**
	a. All Condenser Radiation Values – NORMAL	
8	**Verify Isolation of Ruptured SG(s):**	
	a. Ruptured SG(s) pressure – STABLE OR RISING	
	b. Verify steam dump valve(s) from ruptured SG(s) in CLOSED position	
	c. Check steamline valve(s) from ruptured SG(s) – CLOSED	c. Manually close ruptured SG steamline valve(s)

Step 8 continued on next page.

FOLDOUT PAGE FOR E-2

1. SI REINITIATION CRITERIA

   Manually stop SI pumps if RCS core exit temperature – LESS THAN 25°

2. MULTIPLE FAULTED STEAM GENERATOR CRITERIA

   Stabilize the plant by returning to E-2, FAULTED STEAM GENERATOR, Step 1, if any intact SG level falls in an uncontrolled manner or any intact SG has abnormal radiation.

3. CONTINUOUS ACTION STEPS

   Identify Faulted SG(s) (Step 2)

   Check Secondary Radiation (Step 7)

   Check if SI should be terminated (Step 10)

STEP	ACTION / EXPECTED RESPONSE	RESPONSE NOT OBTAINED
	Step 8 continued	
	d. Check secondary coolant pump(s) from ruptured SG(s) – STOPPED	c. Manually stop ruptured SG secondary coolant pump(s)
	e. Verify RCP from ruptured SG(s) – STOPPED	d. Manually stop ruptured SG RCP(s)
9	**Check RCS Pressure - RISING**	Activate SI Train A and B OR Continue SI
10	**Check if SI Flow Should be Terminated:**	
	a. RCS Pressure – STABLE OR RISING	a. Continue SI
	b. At least ONE intact SG coolant level – GREATER THAN 7%	
11	**Stop Both SI Pumps**	
12	**Check RCP Status:**	
	a. Verify RCP(s) to intact SG(s) are running	a. Manually start RCP(s) for intact SG(s)
	b. Verify RCS Pressure and Temperature – NORMAL	

FOLDOUT PAGE FOR E-2

1. SI REINITIATION CRITERIA

   Manually stop SI pumps if RCS core exit temperature – LESS THAN 25°

2. MULTIPLE FAULTED STEAM GENERATOR CRITERIA

   Stabilize the plant by returning to E-2, FAULTED STEAM GENERATOR, Step 1, if any intact SG level falls in an uncontrolled manner or any intact SG has abnormal radiation.

3. CONTINUOUS ACTION STEPS

   Identify Faulted SG(s) (Step 2)

   Check Secondary Radiation (Step 7)

   Check if SI should be terminated (Step 10)

STEP	ACTION / EXPECTED RESPONSE	RESPONSE NOT OBTAINED
13	**Initiate Rod Extraction:**	
	a. Place rod extraction control in AUTO	
	b. Set rod extraction rate – 10 STEPS PER MINUTE	
	c. Verify rod extraction	c. Manually extract rods – 10 STEPS PER MINUTE
14	**Initiate Power Production:**	
	a. Verify intact SG(s) Temperature – NORMAL	a. Verify intact SG(s) RCPs – ACTIVATED
	b. Intact SG(s) Steamline Valves – OPEN	b. Open intact SG(s) Steamline Valves
	c. Intact SG(s) Secondary Coolant Pumps – ACTIVATED	c. Activate intact SG(s) Secondary Coolant Pumps
	d. Place Steam Dump Valves in CLOSED position	
	e. ACTIVATE turbine(s) for intact loop(s) – Indicators LIT GREEN	

- END –

Number	Title	Rev./Date
E-3	Steam Generator Tube Rupture	Rev.1

Purpose

This procedure provides actions to terminate leakage of reactor coolant into the secondary system following a steam generator tube rupture.

Symptoms or Entry Conditions

This procedure is entered from:

1. E-0, REACTOR SCRAM OR SAFETY INJECTION, Step 9, when condenser radiation is abnormal.
2. E-1, LOSS OF REACTOR OR SECONDARY COOLANT, Step 3 or Step 4
3. E-2, FAULTED STEAM GENERATOR ISOLATION, Step 7

FOLDOUT PAGE FOR E-3

1. SI REINITIATION CRITERIA

   Manually stop SI pumps if RCS core exit temperature – LESS THAN 25° F

2. MULTIPLE TUBE RUPTURE CRITERIA

   Stabilize the plant by returning to E-3, STEAM GENERATOR TUBE RUPTURE, Step 1, if any intact SG level rises in an uncontrolled manner or any intact SG has abnormal radiation.

3. CONTINUOUS ACTION STEPS

   Check if RCPs should be stopped (Step 1)

   Check Ruptured SG(s) Coolant Level (Step 4)

   Check Intact SG(s) Levels (Step 7)

STEP	ACTION / EXPECTED RESPONSE	RESPONSE NOT OBTAINED
1	**Check if RCPs should be stopped:**	
	a. Check RCPs – ANY RUNNING	a. Verify steam dumps in steam pressure mode. Go to Step 2
	b. RCS pressure – LESS THAN 1360 psig	b. Go to Step 2
	c. Stop all RCPs	
	d. Place steam dumps in steam pressure mode	
2	**Identify Ruptured SG(s):**	
	a. High radiation from any SG OR b. High radiation from any SG steamline	

CAUTION: *At least one SG must be maintained available for RCS cooldown.*

3	**Isolate Flow from Ruptured SG(s):**	
	a. Place steam dump valve(s) from ruptured SG(s) in CLOSED position	
	b. Check steamline valve(s) from ruptured SG(s) – CLOSED	b. Manually close ruptured SG steamline valve(s)
	c. Check secondary coolant pump(s) from ruptured SG(s) – STOPPED	c. Manually stop ruptured SG secondary coolant pump(s)
	d. Verify RCP from ruptured SG(s) – STOPPED	d. Manually stop ruptured SG RCP(s)

FOLDOUT PAGE FOR E-3

1. SI REINITIATION CRITERIA

   Manually stop SI pumps if RCS core exit temperature – LESS THAN 25° F

2. MULTIPLE TUBE RUPTURE CRITERIA

   Stabilize the plant by returning to E-3, STEAM GENERATOR TUBE RUPTURE, Step 1, if any intact SG level rises in an uncontrolled manner or any intact SG has abnormal radiation.

3. CONTINUOUS ACTION STEPS

   Check if RCPs should be stopped (Step 1)

   Check Ruptured SG(s) Coolant Level (Step 4)

   Check Intact SG(s) Levels (Step 7)

STEP	ACTION / EXPECTED RESPONSE	RESPONSE NOT OBTAINED
4	**Check Ruptured SG(s) Coolant Level:**	
	a. Coolant Level – GREATER THAN 7%	a. IF coolant level LESS THAN 7%, THEN engage secondary coolant pump until SG level greater than 7%
	b. Verify coolant level - STABLE	
5	**Check Ruptured SG(s) Pressure – GREATER THAN 270 PSIG**	Go to **E-0, REACTOR SCRAM OR SAFETY INJECTION, Step 1**
6	**Initiate RCS Cooldown:**	
	a. Determine required core exit temperature:	

LOWEST RUPTURED SG PRESSURE (PSIG)	CORE EXIT TEMPERATURE (°F)
1000 – 1200	510°
800 – 1000	470°
600 – 800	439°
400 – 600	398°
270 – 400	363°

	b. Dump steam to condenser from all intact SG(s)	
	c. Intact SG Temperature(s) – LESS THAN REQUIRED TEMPERATURE	c. Continue to Step 7
	d. Maintain core exit temperature – LESS THAN REQUIRED TEMPERATURE	d. Dump steam as needed to maintain core exit temperature less than required temperature

FOLDOUT PAGE FOR E-3

1. SI REINITIATION CRITERIA

   Manually stop SI pumps if RCS core exit temperature – LESS THAN 25° F

2. MULTIPLE TUBE RUPTURE CRITERIA

   Stabilize the plant by returning to E-3, STEAM GENERATOR TUBE RUPTURE, Step 1, if any intact SG level rises in an uncontrolled manner or any intact SG has abnormal radiation.

3. CONTINUOUS ACTION STEPS

   Check if RCPs should be stopped (Step 1)

   Check Ruptured SG(s) Coolant Level (Step 4)

   Check Intact SG(s) Levels (Step 7)

STEP	ACTION / EXPECTED RESPONSE	RESPONSE NOT OBTAINED
7	**Check intact SG Levels:**	
	a. Coolant Level – GREATER THAN 7%	a. IF coolant level LESS THAN 7%, THEN engage secondary coolant pump until SG level greater than 7%
	b. Control secondary coolant pump to maintain level between 7% and 70%	b. IF level in any intact SG continues to rise in an uncontrolled manner, THEN return to Step 1
8	**Reset SI Train A and B**	
9	**Verify Isolation of Ruptured SG(s):**	
	a. Ruptured SG(s) pressure – WITHIN LIMITS	
	b. Verify steam dump valve(s) from ruptured SG(s) in CLOSED position	
	c. Check steamline valve(s) from ruptured SG(s) – CLOSED	c. Manually close ruptured SG steamline valve(s)
	d. Check secondary coolant pump(s) from ruptured SG(s) – STOPPED	d. Manually stop ruptured SG secondary coolant pump(s)
	e. Verify RCP from ruptured SG(s) – STOPPED	e. Manually stop ruptured SG RCP(s)
10	**Check RCS Pressure - RISING**	Activate SI Train A and B OR Continue SI

FOLDOUT PAGE FOR E-3

1. SI REINITIATION CRITERIA

   Manually stop SI pumps if RCS core exit temperature – LESS THAN 25° F

2. MULTIPLE TUBE RUPTURE CRITERIA

   Stabilize the plant by returning to E-3, STEAM GENERATOR TUBE RUPTURE, Step 1, if any intact SG level rises in an uncontrolled manner or any intact SG has abnormal radiation.

3. CONTINUOUS ACTION STEPS

   Check if RCPs should be stopped (Step 1)

   Check Ruptured SG(s) Coolant Level (Step 4)

   Check Intact SG(s) Levels (Step 7)

STEP	ACTION / EXPECTED RESPONSE	RESPONSE NOT OBTAINED
11	**Check if SI Flow Should be Terminated:** a. RCS Pressure – STABLE OR RISING	a. Continue SI
	b. At least ONE intact SG coolant level – GREATER THAN 7%	
12	**Stop Both SI Pumps**	
13	**Check RCP Status:**	
	a. Verify RCP(s) to intact SG(s) are running	a. Manually start RCP(s) for intact SG(s)
	b. Verify RCS Pressure and Temperature – NORMAL	
14	**Initiate Rod Extraction:**	
	a. Place rod extraction control in AUTO	
	b. Set rod extraction rate – 10 STEPS PER MINUTE	
	c. Verify rod extraction	c. Manually extract rods – 10 STEPS PER MINUTE
15	**Initiate Power Production:**	
	a. Verify intact SG(s) Temperature – NORMAL	a. Verify intact SG(s) RCPs – ACTIVATED
	b. Intact SG(s) Steamline Valves – OPEN	b. Open intact SG(s) Steamline Valves
	c. Intact SG(s) Secondary Coolant Pumps - ACTIVATED	c. Activate intact SG(s) Secondary Coolant Pumps

Step 15 continued on next page

FOLDOUT PAGE FOR E-3

1. SI REINITIATION CRITERIA

   Manually stop SI pumps if RCS core exit temperature – LESS THAN 25° F

2. MULTIPLE TUBE RUPTURE CRITERIA

   Stabilize the plant by returning to E-3, STEAM GENERATOR TUBE RUPTURE, Step 1, if any intact SG level rises in an uncontrolled manner or any intact SG has abnormal radiation.

3. CONTINUOUS ACTION STEPS

   Check if RCPs should be stopped (Step 1)

   Check Ruptured SG(s) Coolant Level (Step 4)

   Check Intact SG(s) Levels (Step 7)

STEP	ACTION / EXPECTED RESPONSE	RESPONSE NOT OBTAINED

Step 15 continued

d. Place Steam Dump Valves in CLOSED position

e. ACTIVATE turbine(s) for intact loop(s) – Indicators LIT GREEN

- END -

A. PURPOSE

This Anomalous Operating Procedure provides actions to rectify loop malfunctions in case of an unexpected power excursion.

B. SYMPTOMS OR ENTRY CONDITIONS

Condition	Description
1	Check Condenser Radiation Level – RISING
2	Check SG Temperature – RISING
2	Check Turbine Steam Pressure – RISING
4	Check Cooling Tower Temp – RISING
5	Check Power Output – GREATER THAN 10 MW

Step	Action/Expected Response	Response Not Obtained
1	**IF Power Output – GREATER THAN 12 MW THEN Contact Generator Maintenance Department – ERROR CODE: AOP-1A AND Halt Power Production for leaking loop ONLY** -ALL RCPs to affected SG – DEACTIVATED -Verify Power Output – DROPPING -Verify Turbine Steam Press – DROPPING -Verify SG Temperature – DROPPING -Verify SG Pressure – DROPPING -Affected SGs Steamline Valves – CLOSED -Affected SGs Secondary Coolant Pumps – DEACTIVATED -Deactivate Turbines for affected loop – indicators LIT RED -Verify Expected Alarm – Turbine Auto Trip LIT RED	-Deactivate RCPs to affected SG -Close SG Steamline Valves -Deactivate SG Secondary Coolant Pumps
2	**IF Power Output – LESS THAN 12 MW THEN Identify affected loop as undergoing MINOR POWER EXCURSION** -RCP for affected loop – DEACTIVATED -Verify Condenser Radiation Level – STABLE OR DROPPING -Verify SG Temperature – STABLE OR DROPPING -Verify Turbine Steam Pressure – STABLE OR DROPPING -Verify Coolant Tower Temperature – STABLE OR DROPPING -Verify Power Output – DROPPING	-Deactivate RCP for affected loop -Contact Generator Maintenance Department – ERROR CODE: AOP-1B

A. PURPOSE

This Anomalous Operating Procedure provides actions to rectify the unexpected loss of data feeds from reactor cooling towers.

B. SYMPTOMS OR ENTRY CONDITIONS

Condition	Description
1	Check one or more loop status indicators – NONFUCTIONAL

Step	Action/Expected Response	Response Not Obtained
1	**Contact Generator Maintenance Department – ERROR CODE: AOP-2**	

A. PURPOSE

This Anomalous Operating Procedure provides actions to rectify the unexpected loss of data feeds from the reactor.

B. SYMPTOMS OR ENTRY CONDITIONS

Condition	Description
1	Check one or more reactor status indicators – NONFUCTIONAL

Step	Action/Expected Response	Response Not Obtained
1	**Contact Reactor Maintenance Department – ERROR CODE: AOP-2**	
2	**Contact Containment Building Management – ERROR CODE: ASO-2001**	
3	**IF data loss is in Reactor Temperature, Containment Radiation, Containment Pressure, or Containment Temperature indicators, THEN lower rods 50%** -Rod insertion control – AUTO -Rod insertion rate – 25 steps/min -Rod insertion control – AUTO OFF when rod insertion is at 50% -Verify rod insertion	-Manually insert rods – 25 steps/min -Stop inserting when insertion is at 50%

A. PURPOSE

This Anomalous Operating Procedure provides actions to rectify the unexpected loss of data feeds from the rod sensors.

B. SYMPTOMS OR ENTRY CONDITIONS

Condition	Description
1	Check rod insertion control – AUTO
2	Check rod insertion display – NONFUNCTIONAL

Step	Action/Expected Response	Response Not Obtained
1	**Rod insertion control – AUTO OFF**	
2	**Rod insertion control – AUTO ON**	
3	**Verify rod insertion display – FUNCTIONAL**	-Contact Reactor Maintenance Department – ERROR CODE: ISD-424238

A. PURPOSE

This Anomalous Operating Procedure provides actions in the event of a malfunction of Turbine Power Generation

B. SYMPTOMS OR ENTRY CONDITIONS

Condition	Description
1	Check SG Pressure – STABLE OR RISING
2	Check SG Temperature – STABLE OR RISING
3	Check Turbine Steam Pressure – STABLE OR RISING
4	Check Power Output – DROPPING

Step	Action/Expected Response	Response Not Obtained
1	**IDENTIFY FAULT AS FAULTED TURBINE IN AFFECTED LOOP**	
2	**Halt Power Production for affected loop ONLY**	
	-ALL RCPs to affected SG – DEACTIVATED	-Deactivate RCPs to affected SG
	-Verify Power Output – DROPPING	
	-Verify Turbine Steam Press – DROPPING	
	-Verify SG Temperature – DROPPING	
	-Verify SG Pressure – DROPPING	
	-Affected SGs Steamline Valves – CLOSED	-Close SG Steamline Valves
	-Affected SGs Secondary Coolant Pumps – DEACTIVATED	-Deactivate SG Secondary Coolant Pumps
	-Deactivate Turbines for affected loop – indicators LIT RED	
	-Verify Expected Alarm – Turbine Auto Trip LIT RED	
3	**Contact Reactor Maintenance Department – ERROR CODE: AOP-5**	

A. PURPOSE

This Anomalous Operating Procedure provides actions to diagnose and rectify anomalies in reactor cooling loop temperature or pressure.

B. SYMPTOMS OR ENTRY CONDITIONS

Condition	Description
1	Check SG Pressure – DROPPING
2	Check SG Temperature – DROPPING
2	Check Turbine Steam Pressure – DROPPING
4	Check Power Output – DROPPING

Number	Title	Rev./Date
AOP-6	Loop Temperature/Pressure Anomaly	Rev.6 6/17/2011

Step	Action/Expected Response	Response Not Obtained
1	**Check Reactor Coolant Pump** -Deactivate RCP in affected loop -Activate RCP in affected loop -Verify SG Pressure – RISING -Verify SG Temperature – RISING -Verify Turbine Steam Pressure – RISING -Verify Power Output – RISING -IF the conditions have been verified, THEN IDENTIFY FAULT AS FAULTED REACTOR COOLANT PUMP AND DO NOT CONTINUE THIS AOP	-Continue on to next step
2	**Check Steam Dump Valve** -Open Steam Dump Valve in affected loop -Close Steam Dump Valve in affected loop -Verify SG Pressure –RISING -Verify SG Temperature –RISING -Verify Turbine Steam Pressure –RISING -Verify Power Output –RISING -IF the conditions have been verified, THEN IDENTIFY FAULT AS FAULTED STEAM DUMP VALVE AND DO NOT CONTINUE THIS AOP	-IDENFITY FAULT AS STEAM DUMP VALVE LEAK AND CONTINUE TO HALT POWER PRODUCTION STEP
3	**Halt Power Production for leaking loop ONLY** -Deactivate all RCPs to affected SGs -Verify Power Output – DROPPING -Verify Turbine Steam Press – DROPPING -Verify SG Temperature – DROPPING -Verify SG Pressure – DROPPING -Close affected SGs Steamline Valves -Deactivate affected SGs Secondary Coolant Pumps – -Deactivate Turbines for affected loop – indicators LIT RED	
4	**Contact Reactor Maintenance Department – ERROR CODE: AOP-6**	

Appendix	Title	Rev./Date
A	Normal Operating Meter Values	Rev.1 7/18/2011

A. PURPOSE

This appendix provides guidelines for the values that would be considered normal during optimal nuclear reactor operations. While these values are present on meters, the limit indicator will show them as being normal (a thin dash). If meter values are below the lower limit, a thick dash will appear. If meter values are above the upper limit, a plus sign (+) will appear.

Meter values follow their corresponding units.

B. Normal Operating Meter Values

Meter Type	Units	Range
Reactor Temperature	°F	350-550
RCS Temperature	°F	350-550
RCS Pressure	PSIG	250-1300
Coolant Level	%	40-85
Neutron Flux	mrad	100-981
Containment Radiation	mrad	0-10
Containment Pressure	PSIG	20-30
Containment Temperature	°F	60-110
Condenser Radiation Level	mrad	0-2
Secondary Flow Rate	GPM	600-1200
Cooling Tower Temperature	°F	40-100
Coolant Temperature	°F	70-250
SG Temperature	°F	350-550
SG Coolant Level	%	15-45
SG Pressure	PSIG	250-1300
Turbine Steam Pressure	PSIG	250-1000
Power Output	MW	0-10

A. PURPOSE

This appendix provides an overview of power plant departments that a reactor operator may need to contact. A description of each department's role, as well as their abbreviations will be provided.

B. Relevant Power Plant Departments

Reactor Maintenance (RM)

The Reactor Maintenance Department is in charge of ensuring the normal operations of the nuclear reactor core, including handling of fissile material, ensuring normal flow of working fluid to and from the reactor, and maintaining the steam generators.

Generator Management (GM)

The Generator Management Department is in charge of the power generation portion of the plant. This department deals with any problems pertaining to the turbines, cooling towers, and related plant components.

Containment Building (CB)

The Containment Building Department manages the shielding and isolation of the reactor core from the outside environment. Safety considerations regarding internal and external breaches to the containment structure are handled by this department.

Appendix F

Data Transformations in Rule-Based Environment

This appendix presents the analysis for selecting the appropriate data transformation for the rule-based dataset using a k-means algorithm. In addition to a subjective assessment of the clustering results, k-means clustering using each range and z-score transformations were compared using the external metrics of entropy and purity, and the internal metrics of compactness and isolation. For both external metrics, performance was similar across both transformations and thus provided little evidence towards selecting a tranformation. Internal metrics (compactness and isolation) provided more variation across the two transformations. However, due to the different scaling factors they could not be directly compared between the range and z-score transformations.

It is understood that a high average inter-centroid distance (isolation) and low SSE (compactness) are desirable, and therefore the ratio of isolation to compactness was used to compare the transformations. The results are presented in Table F.1. In the table, the z-score transformation performs better (as determined by a higher isolation to compactness ratio) for adherence features, while the range transformation performs better for all other metrics. A single transformation should be used for all features, and thus it must be determined which feature sets are more "important" for the clustering algorithms. In this case, the adherence metrics proposed here are not currently used in rule-based settings, while the other metrics are common in current training programs. Therefore, it seems appropriate to select the range transformation which provides the best utility for currently gathered assessment metrics.

Table F.1: Comparison of Isolation to Compactness Ratio for Range and Z-Score Transformations.

Features	Range Transformation	Z-Score Transformation
Levenshtein Distance	0.894	1.059
PAM	0.923	1.019
Objective Metrics	0.206	0.161
Subjective Metrics	0.248	0.202
Targets	1.138	0.67

Appendix G

Calculation of BIC

This appendix discusses the calcuation of the Bayesian Information Criterion (BIC), a common method for the selection of model parameters to minimize both the model complexity and model error. BIC has been shown to be useful in the selection of the number of clusters in K-means [147], and a similar method was implemented in this research. The basic formula for BIC is given by Kass and Wasserman [173], and is reproduced as Equation G.1. This and the following equations assume given the data D, and a family of models M_j which correspond to different numbers of clusters K which have R_j data points assigned to that cluster.

$$BIC(M_j) = \hat{l}_j(D) - \frac{p_j}{2} \cdot logR \tag{G.1}$$

Under the identical spherical Gaussian assumption, the maximum likelihood estimate (MLE) of the variance is given in Equation G.2.

$$\hat{\sigma}^2 = \frac{1}{R-K} \sum_i (x_i - \mu_{(i)})^2 \tag{G.2}$$

Just focusing on the set of points D_n which belong to centroid n, the estimate of the log-likelihood for that centroid can be found as given in Equation G.3.

$$\hat{l}(D_n) = -\frac{R_n}{2} log(2\pi) - \frac{R_n \cdot M}{2} log(\hat{\sigma}^2) - \frac{R_n - K}{2} + R_n logR_n - R_n logR \tag{G.3}$$

Calculating Equation G.3 for each centroid (cluster) and summing over these likeli-

hood values provides the estimate for the total likelihood of the model $\hat{l}_j(D)$ in Equation G.1. The number of free parameters p_j for each model is found as the sum of $K-1$ class probabilities, $M \cdot K$ centroid coordinates, and a single variance estimate. With R representing the number of data points in the entire dataset, Equation G.1 can then be used to calculate the BIC for that particular K-means model.

Appendix H

Example Cluster Assignments from K-means and Hierarchical Clustering

This appendix provides an example set of cluster assignments from k-means and hierarchical clustering on the rule-based dataset. Table H.1 shows the cluster assignments for k-means and hierarchical clustering for the 47 participants for a single run of the algorithm using levenshtein distance. It can be seen that hierarchical clustering almost exclusively favors putting the trainees into a single cluster, with only one trainee in each cluster 1 and cluster 2. On the other hand, k-means, provides a much more even split between clusters across the trainees in the dataset.

The assignment of trainees into a single cluster creates significant challenges for a training supervisor. With most or all trainees grouped together, it is difficult to select a subset of trainees to apply a training intervention (TI). Therefore, while hierarchical clustering provides other advantages compared to k-means (such as measured by purity), the resultant cluster assignments are not useful from an intervention perspective. Given this, it is preferable to utilize k-means over hierarchical clustering on this dataset.

Table H.1: Comparison of Cluster Assignments for K-means and Hierarchical Clustering on Rule-based Dataset

Trainee	K-means	Hierarchical Clustering
1	2	3
2	1	1
3	3	3
4	3	3
5	3	3
6	1	3
7	3	3
8	2	3
9	3	3
10	2	3
11	3	3
12	2	3
13	2	3
14	2	3
15	1	2
16	1	3
17	1	3
18	1	3
19	1	3
20	1	3
21	1	3
22	3	3
23	2	3
24	3	3
25	2	3
26	2	3
27	2	3
28	3	3
29	2	3
30	2	3
31	2	3
32	2	3
33	3	3
34	2	3
35	2	3
36	2	3
37	1	3
38	2	3
39	2	3
40	2	3
41	2	3
42	2	3
43	2	3
44	2	3
45	2	3
46	3	3
47	2	3

Appendix I

PCA Components from Rule-Based Dataset

Table I.1: PCA Components from Rule-Based Dataset

Module	Action	Component 1	Component 2	Component 3
1	1	9.22E-05	0.001554	-0.00241
1	2	0.000604	0.000883	-0.00739
1	3	0.001058	0.00166	-0.01217
1	4	0.001662	0.003144	-0.01844
1	5	0.002453	0.00513	-0.02626
1	6	0.003184	0.007121	-0.0345
1	7	0.003887	0.008881	-0.04253
1	8	0.004566	0.010818	-0.05023
1	9	0.005306	0.013389	-0.05367
1	10	0.006068	0.015858	-0.05692
1	11	0.006865	0.018331	-0.06346
1	12	0.007641	0.019984	-0.06623
1	13	0.008318	0.022023	-0.07329
1	14	0.009132	0.025555	-0.07938
1	15	0.010014	0.028497	-0.08273
1	16	0.010887	0.031012	-0.08796

Continued on next page

Table I.1 – continued from previous page

Module	Action	Component 1	Component 2	Component 3
1	17	0.011763	0.033976	-0.09025
1	18	0.012627	0.037574	-0.09258
1	19	0.01349	0.040639	-0.0965
1	20	0.014264	0.042727	-0.10021
1	21	0.014962	0.044305	-0.10132
1	22	0.015664	0.0462	-0.10425
1	23	0.016503	0.048648	-0.10846
1	24	0.017403	0.050123	-0.10736
1	25	0.018272	0.05173	-0.10715
1	26	0.019072	0.053303	-0.10444
1	27	0.019953	0.056378	-0.10449
1	28	0.020878	0.059143	-0.107
1	29	0.021808	0.06154	-0.10541
1	30	0.022719	0.064489	-0.10318
1	31	0.023607	0.067231	-0.10392
1	32	0.024523	0.069748	-0.10329
1	33	0.025394	0.072571	-0.10455
1	34	0.026209	0.075491	-0.10614
1	35	0.027092	0.078726	-0.10517
1	36	0.027825	0.081208	-0.10902
1	37	0.028563	0.084015	-0.11297
1	38	0.029337	0.087668	-0.11475
1	39	0.030068	0.091062	-0.11702
1	40	0.030758	0.093271	-0.11613
1	41	0.031483	0.095973	-0.11815
1	42	0.032175	0.099393	-0.12017
1	43	0.033005	0.102181	-0.11764
1	44	0.033872	0.104076	-0.11361
1	45	0.034623	0.106765	-0.11218

Continued on next page

Table I.1 – continued from previous page

Module	Action	Component 1	Component 2	Component 3
1	46	0.035383	0.109118	-0.10923
1	47	0.03616	0.111115	-0.1078
1	48	0.036932	0.111857	-0.10395
1	49	0.037827	0.113679	-0.10114
1	50	0.038674	0.115778	-0.09872
1	51	0.039326	0.116598	-0.08912
1	52	0.039903	0.116014	-0.08377
1	53	0.04071	0.116924	-0.08076
1	54	0.041567	0.119236	-0.08006
1	55	0.042225	0.1209	-0.07259
1	56	0.042783	0.121327	-0.06562
1	57	0.043412	0.121863	-0.05978
1	58	0.044077	0.123304	-0.05514
1	59	0.044763	0.12484	-0.05018
1	60	0.045574	0.125797	-0.04937
1	61	0.046397	0.125852	-0.04644
1	62	0.047035	0.126921	-0.0418
1	63	0.047645	0.128371	-0.03768
1	64	0.048439	0.131199	-0.03236
1	65	0.04932	0.132275	-0.02811
1	66	0.050097	0.13186	-0.0219
1	67	0.050763	0.130861	-0.01315
1	68	0.051365	0.130836	-0.00248
1	69	0.052144	0.128649	0.006625
1	70	0.052642	0.124336	0.018473
1	71	0.053179	0.120561	0.030359
1	72	0.053692	0.116756	0.041494
1	73	0.054405	0.111775	0.051913
1	74	0.05514	0.107923	0.061834

Continued on next page

Table I.1 – continued from previous page

Module	Action	Component 1	Component 2	Component 3
1	75	0.055899	0.10342	0.074072
1	76	0.056798	0.104535	0.077005
1	77	0.057147	0.101189	0.081178
1	78	0.057548	0.099763	0.085204
1	79	0.058208	0.097667	0.094164
1	80	0.058551	0.097098	0.102223
1	81	0.059075	0.094878	0.10716
1	82	0.059435	0.092857	0.114369
1	83	0.059966	0.09176	0.122807
1	84	0.060375	0.08975	0.125497
1	85	0.060747	0.088031	0.129928
1	86	0.061196	0.08594	0.130737
1	87	0.06158	0.083229	0.131568
1	88	0.061897	0.080893	0.130726
1	89	0.061877	0.080424	0.129872
1	90	0.062077	0.077933	0.127525
1	91	0.062276	0.075442	0.125177
1	92	0.06254	0.073108	0.124798
1	93	0.062803	0.070775	0.12442
1	94	0.062949	0.068286	0.122535
1	95	0.063286	0.06642	0.122548
1	96	0.063506	0.064399	0.121054
1	97	0.063726	0.062377	0.11956
1	98	0.064066	0.059672	0.115559
1	99	0.064286	0.057651	0.114065
1	100	0.064563	0.054788	0.108094
1	101	0.064968	0.052239	0.106063
1	102	0.065309	0.049534	0.102061
1	103	0.065649	0.046828	0.09806

Continued on next page

Table I.1 – continued from previous page

Module	Action	Component 1	Component 2	Component 3
1	104	0.06599	0.044122	0.094058
1	105	0.066331	0.041416	0.090057
1	106	0.066672	0.038711	0.086056
1	107	0.067013	0.036005	0.082054
1	108	0.067134	0.03532	0.079547
1	109	0.067255	0.034636	0.077039
1	110	0.067596	0.03193	0.073038
1	111	0.067937	0.029224	0.069037
1	112	0.068278	0.026519	0.065035
1	113	0.068619	0.023813	0.061034
1	114	0.068959	0.021107	0.057033
1	115	0.0693	0.018402	0.053031
1	116	0.069421	0.017717	0.050524
1	117	0.069762	0.015011	0.046522
1	118	0.070103	0.012306	0.042521
1	119	0.070444	0.0096	0.03852
1	120	0.070565	0.008915	0.036012
1	121	0.070906	0.00621	0.032011
1	122	0.071247	0.003504	0.028009
1	123	0.071588	0.000798	0.024008
1	124	0.071928	-0.00191	0.020007
1	125	0.072269	-0.00461	0.016005
1	126	0.07261	-0.00732	0.012004
1	127	0.072951	-0.01002	0.008003
1	128	0.073171	-0.01205	0.006509
1	129	0.073171	-0.01205	0.006509
1	130	0.07339	-0.01407	0.005015
1	131	0.073731	-0.01677	0.001014
1	132	0.073731	-0.01677	0.001014

Continued on next page

Table I.1 – continued from previous page

Module	Action	Component 1	Component 2	Component 3
1	133	0.074072	-0.01948	-0.00299
1	134	0.074193	-0.02016	-0.0055
1	135	0.074292	-0.0215	-0.00448
1	136	0.07439	-0.02284	-0.00347
1	137	0.07461	-0.02486	-0.00496
1	138	0.074829	-0.02688	-0.00646
1	139	0.074829	-0.02688	-0.00646
1	140	0.075049	-0.0289	-0.00795
1	141	0.075269	-0.03092	-0.00944
1	142	0.075488	-0.03294	-0.01094
1	143	0.075488	-0.03294	-0.01094
1	144	0.075708	-0.03496	-0.01243
1	145	0.075928	-0.03698	-0.01392
1	146	0.076147	-0.039	-0.01542
1	147	0.076367	-0.04103	-0.01691
1	148	0.076587	-0.04305	-0.01841
1	149	0.076806	-0.04507	-0.0199
2	1	0.000439	0.003838	-0.00712
2	2	0.00132	0.008886	-0.00647
2	3	0.002224	0.014181	-0.00425
2	4	0.003265	0.019409	-0.00144
2	5	0.004422	0.027569	0.004847
2	6	0.005567	0.035291	0.013983
2	7	0.006743	0.042174	0.024004
2	8	0.007915	0.049038	0.033598
2	9	0.008631	0.052284	0.040373
2	10	0.009147	0.05438	0.047662
2	11	0.010026	0.060314	0.061525
2	12	0.010859	0.06491	0.074479

Continued on next page

Table I.1 – continued from previous page

Module	Action	Component 1	Component 2	Component 3
2	13	0.011912	0.070767	0.083264
2	14	0.013088	0.076848	0.092088
2	15	0.014189	0.082205	0.102794
2	16	0.015156	0.087791	0.110455
2	17	0.015834	0.091365	0.117798
2	18	0.016633	0.09574	0.12061
2	19	0.017591	0.102017	0.125515
2	20	0.018302	0.108943	0.140611
2	21	0.01917	0.1124	0.145754
2	22	0.019959	0.114706	0.148276
2	23	0.020793	0.116444	0.147356
2	24	0.021847	0.120989	0.150147
2	25	0.022988	0.125919	0.151068
2	26	0.023975	0.1313	0.14934
2	27	0.024902	0.135776	0.146525
2	28	0.025915	0.139576	0.143132
2	29	0.026897	0.143688	0.142154
2	30	0.028032	0.149462	0.140379
2	31	0.029175	0.153909	0.133257
2	32	0.030189	0.157794	0.128766
2	33	0.031061	0.160725	0.122409
2	34	0.032125	0.164396	0.109939
2	35	0.03346	0.169641	0.099961
2	36	0.03395	0.167655	0.071078
2	37	0.035018	0.164882	0.054587
2	38	0.035534	0.16485	0.034416
2	39	0.035767	0.163453	0.021352
2	40	0.03658	0.158125	0.008218
2	41	0.037463	0.153765	-0.00514

Continued on next page

Table I.1 – continued from previous page

Module	Action	Component 1	Component 2	Component 3
2	42	0.038218	0.149251	-0.01548
2	43	0.038959	0.144631	-0.02963
2	44	0.039615	0.138327	-0.04415
2	45	0.040314	0.13288	-0.05361
2	46	0.041173	0.129945	-0.06476
2	47	0.041651	0.13107	-0.07966
2	48	0.042299	0.127347	-0.0787
2	49	0.043094	0.125797	-0.07752
2	50	0.043888	0.124248	-0.07634
2	51	0.044646	0.122244	-0.07515
2	52	0.045403	0.12024	-0.07396
2	53	0.046094	0.116376	-0.07942
2	54	0.046858	0.113147	-0.08549
2	55	0.047622	0.109919	-0.09156
2	56	0.048313	0.106054	-0.09702
2	57	0.048919	0.100507	-0.10285
2	58	0.049596	0.096304	-0.10639
2	59	0.050273	0.0921	-0.10993
2	60	0.05095	0.087897	-0.11347
2	61	0.051627	0.083694	-0.117
2	62	0.052303	0.079491	-0.12054
2	63	0.053065	0.076971	-0.12371
2	64	0.053827	0.074451	-0.12688
2	65	0.054588	0.071932	-0.13005
2	66	0.05535	0.069412	-0.13321
2	67	0.056111	0.066892	-0.13638
2	68	0.056873	0.064372	-0.13955
2	69	0.057634	0.061852	-0.14272
2	70	0.058396	0.059332	-0.14589

Continued on next page

Table I.1 – continued from previous page

Module	Action	Component 1	Component 2	Component 3
2	71	0.059157	0.056812	-0.14905
2	72	0.058844	0.056723	-0.14061
2	73	0.05898	0.054024	-0.12689
2	74	0.059741	0.051504	-0.13006
2	75	0.060503	0.048984	-0.13323
2	76	0.060951	0.046375	-0.12796
2	77	0.061399	0.043766	-0.12268
2	78	0.061848	0.041157	-0.11741
2	79	0.062296	0.038547	-0.11213
2	80	0.062745	0.035938	-0.10686
2	81	0.063193	0.033329	-0.10158
2	82	0.063642	0.030719	-0.09631
2	83	0.06409	0.02811	-0.09103
2	84	0.064538	0.025501	-0.08576
2	85	0.064987	0.022892	-0.08048
2	86	0.065435	0.020282	-0.07521
2	87	0.065884	0.017673	-0.06994
2	88	0.066332	0.015064	-0.06466
2	89	0.066781	0.012455	-0.05939
2	90	0.067229	0.009845	-0.05411
2	91	0.067677	0.007236	-0.04884
2	92	0.068126	0.004627	-0.04356
2	93	0.068574	0.002018	-0.03829
2	94	0.069023	-0.00059	-0.03301
2	95	0.069471	-0.0032	-0.02774
2	96	0.069919	-0.00581	-0.02246
2	97	0.070368	-0.00842	-0.01719
2	98	0.070816	-0.01103	-0.01191
2	99	0.071265	-0.01364	-0.00664

Continued on next page

Table I.1 – continued from previous page

Module	Action	Component 1	Component 2	Component 3
2	100	0.071713	-0.01625	-0.00137
2	101	0.072162	-0.01886	0.003909
2	102	0.07261	-0.02147	0.009183
2	103	0.073058	-0.02408	0.014458
2	104	0.073507	-0.02668	0.019733
3	1	2.34E-05	0.004954	0.004429
3	2	0.000297	0.009335	0.010497
3	3	0.000542	0.014247	0.014589
3	4	0.000597	0.019637	0.01988
3	5	0.000853	0.023843	0.025153
3	6	0.001074	0.027944	0.031977
3	7	0.001294	0.034112	0.039721
3	8	0.001472	0.040331	0.046186
3	9	0.001911	0.046955	0.051108
3	10	0.002651	0.051705	0.054524
3	11	0.003226	0.056526	0.059458
3	12	0.003594	0.062579	0.067415
3	13	0.003882	0.068329	0.075526
3	14	0.004081	0.073021	0.084403
3	15	0.004704	0.076877	0.089826
3	16	0.005469	0.081019	0.094583
3	17	0.005738	0.082789	0.094165
3	18	0.00595	0.083694	0.097898
3	19	0.006368	0.08846	0.102237
3	20	0.006916	0.091725	0.10596
3	21	0.007543	0.094548	0.110288
3	22	0.008039	0.098282	0.115701
3	23	0.008295	0.103266	0.120805
3	24	0.008484	0.109328	0.127839

Continued on next page

Table I.1 – continued from previous page

Module	Action	Component 1	Component 2	Component 3
3	25	0.008724	0.115077	0.133512
3	26	0.009017	0.118644	0.139586
3	27	0.009075	0.123075	0.14724
3	28	0.009608	0.128118	0.148652
3	29	0.009994	0.132817	0.151508
3	30	0.010445	0.135451	0.149654
3	31	0.011197	0.137856	0.147781
3	32	0.01192	0.137214	0.143904
3	33	0.012722	0.137085	0.137241
3	34	0.013448	0.136104	0.127658
3	35	0.013934	0.136166	0.119118
3	36	0.014322	0.137388	0.110954
3	37	0.014938	0.137274	0.101875
3	38	0.0159	0.136673	0.089931
3	39	0.016604	0.136652	0.080748
3	40	0.01724	0.136331	0.072477
3	41	0.017905	0.136941	0.063777
3	42	0.018577	0.138217	0.056741
3	43	0.019378	0.138828	0.049729
3	44	0.020222	0.139398	0.041369
3	45	0.021122	0.140076	0.032798
3	46	0.022395	0.139662	0.024015
3	47	0.023696	0.13908	0.015361
3	48	0.025026	0.138548	0.005694
3	49	0.026124	0.137702	-0.00683
3	50	0.027407	0.136242	-0.0164
3	51	0.028585	0.133846	-0.02464
3	52	0.029738	0.130857	-0.03439
3	53	0.030972	0.129202	-0.04394

Continued on next page

Table I.1 – continued from previous page

Module	Action	Component 1	Component 2	Component 3
3	54	0.031982	0.128066	-0.05542
3	55	0.033225	0.126604	-0.06334
3	56	0.034539	0.125672	-0.07289
3	57	0.035541	0.123703	-0.08529
3	58	0.036657	0.120658	-0.09252
3	59	0.037773	0.117613	-0.09974
3	60	0.038915	0.115354	-0.1047
3	61	0.040127	0.11378	-0.11017
3	62	0.041206	0.110688	-0.11422
3	63	0.042284	0.107597	-0.11827
3	64	0.043496	0.105912	-0.12214
3	65	0.044707	0.104227	-0.12602
3	66	0.045807	0.101771	-0.12768
3	67	0.046892	0.099079	-0.12984
3	68	0.048062	0.097308	-0.13202
3	69	0.049232	0.095538	-0.13419
3	70	0.050318	0.093446	-0.13369
3	71	0.05131	0.090873	-0.13205
3	72	0.052192	0.087712	-0.12941
3	73	0.053074	0.084551	-0.12676
3	74	0.053957	0.08139	-0.12412
3	75	0.054839	0.078229	-0.12148
3	76	0.055722	0.075068	-0.11883
3	77	0.056604	0.071906	-0.11619
3	78	0.057486	0.068745	-0.11355
3	79	0.058369	0.065584	-0.1109
3	80	0.059251	0.062423	-0.10826
3	81	0.060134	0.059262	-0.10562
3	82	0.061016	0.056101	-0.10297

Continued on next page

Table I.1 – continued from previous page

Module	Action	Component 1	Component 2	Component 3
3	83	0.061899	0.05294	-0.10033
3	84	0.062781	0.049778	-0.09769
3	85	0.063663	0.046617	-0.09504
3	86	0.064317	0.043754	-0.09505
3	87	0.06497	0.040891	-0.09505
3	88	0.065853	0.03773	-0.09241
3	89	0.06662	0.034593	-0.08939
3	90	0.067387	0.031457	-0.08638
3	91	0.068154	0.028321	-0.08336
3	92	0.068922	0.025184	-0.08034
3	93	0.069689	0.022048	-0.07733
3	94	0.070331	0.019233	-0.07347
3	95	0.070973	0.016418	-0.06962
3	96	0.071615	0.013604	-0.06576
3	97	0.072256	0.010789	-0.06191
3	98	0.072898	0.007974	-0.05805
3	99	0.07354	0.00516	-0.0542
3	100	0.074182	0.002345	-0.05034
3	101	0.074824	-0.00047	-0.04649
3	102	0.07535	-0.00251	-0.04239
3	103	0.075876	-0.00455	-0.03829
3	104	0.076267	-0.0059	-0.03423
3	105	0.076658	-0.00726	-0.03018
3	106	0.077049	-0.00861	-0.02612
3	107	0.07744	-0.00996	-0.02206
3	108	0.077831	-0.01132	-0.01801
3	109	0.078222	-0.01267	-0.01395
3	110	0.078613	-0.01402	-0.0099
3	111	0.079004	-0.01538	-0.00584

Continued on next page

Table I.1 – continued from previous page

Module	Action	Component 1	Component 2	Component 3
3	112	0.079395	-0.01673	-0.00178
3	113	0.079786	-0.01808	0.002272
3	114	0.080177	-0.01944	0.006328
3	115	0.080568	-0.02079	0.010384
3	116	0.080959	-0.02214	0.01444
3	117	0.08135	-0.02349	0.018496
3	118	0.081579	-0.02379	0.021145
3	119	0.081808	-0.02409	0.023794
3	120	0.082037	-0.02439	0.026444

Appendix J

Data Transformations in Knowledge-Based Environment

This appendix presents the analysis for selecting the appropriate data transformation for the knowledge-based dataset using a k-means algorithm. In addition to a subjective assessment of the clustering results, k-means clustering using each range and z-score transformations were compared using the external metrics of entropy and purity, and the internal metrics of compactness and isolation. Across the two tranformations, the z-score transformation had a greater tendency to combine clusters together (such as grouping "A" and "B" together, or "B" and "C" together). For the external metrics performance was similar across both transformations, and thus provided little evidence towards selecting a tranformation. Internal metrics (compactness and isolation) provided more variation across the two transformations. However, due to the different scaling factors they could not be directly compared between the range and z-score transformations.

It is understood that a high average inter-centroid distance (isolation) and low SSE (compactness) are desirable, and therefore the ratio of isolation to compactness was used to compare the transformations. The results are presented in Table ??. It can be seen that the range transformation tends to perform better (as measured by a high ratio) when summative and average information is used, while z-score transformation performs better when individual process-level metrics are available. Given the much greater relative grade contribution of the summative metrics, this indicates that it is desirable to utilize the range transformation to capture this information.

Table J.1: Comparison of Isolation to Compactness Ratio for Range and Z-Score Transformations.

Features	Range Transformation	Z-Score Transformation
Summative Averages	1.069571	0.919112
Summative Individual	0.375329	0.371393
Process-Level Averages	Empty Clusters	Empty Clusters
Process-Level Individual	0.487769	0.83028
Total Averages	0.708137	0.593149
Total Individual	0.272575	0.407062

Appendix K

PCA Results from Knowledge-Based Dataset

This appendix presents the full results of the PCA analysis on the knowledge-based dataset discussed in Chapter 5. The first three principal components are shown in Table K.1. These principle components represent the orthogonal dimensions in which the data show the greatest variability. We can see that in the first component, the major players are primarily quizzes (Quiz 9, Quiz 14, Quiz 13, Quiz 1), though the later more cumulative measures also show some contribution (Problem Set 2, Test 2). In the second principle component, we find similar results, with Quiz 5, Quiz 6, Quiz 17, Quiz 4, and Test 2 being the major elements. The third principle component focuses on Quiz 1 as a major source of variation, and has little other contribution.

Since supervised learning algorithms have an easier time forming decision boundaries or other models for data with higher variance, we can use the PCA results to help inform reasonable choices for inputs into the supervised learning approach. In this case, it is apparent that the quizzes contain considerable variation across the trainees, and may be useful as features for supervised models. Additionally, the tests (particularly test 2) may offer some advantages as a feature as well.

Table K.1: First three principle components on classroom dataset by PCA

Feature	Component 1	Component 2	Component 3
Quiz 1	0.254	-0.143	0.829
Quiz 2	0.115	0.024	0.081
Quiz 3	0.046	0.086	0.007
Quiz 4	-0.084	0.311	-0.177
Quiz 5	0.000	0.615	0.121
Quiz 6	0.009	0.568	0.148
Quiz 7	0.109	-0.012	0.028
Quiz 8	0.198	0.07	-0.051
Quiz 9	0.686	-0.003	-0.217
Quiz 10	0.114	-0.04	-0.19
Quiz 11	0.009	0.037	0.035
Quiz 12	0.039	0.065	0.023
Quiz 13	0.288	-0.016	0.037
Quiz 14	0.372	-0.039	0.08
Quiz 15	0.171	0.071	-0.097
Quiz 16	0.109	0.004	-0.014
Quiz 17	0.049	0.323	0.154
Quiz 18	-0.006	0.069	-0.162
Quiz 19	0.083	0.003	-0.014
Project 1	0.029	0.004	-0.084
Project 2	0.009	-0.035	-0.036
Project 3	0.023	0.006	-0.028
Problem Set 1	0.075	-0.001	-0.036
Problem Set 2	0.227	-0.05	-0.209
Test 1	0.075	0.056	-0.078
Test 2	0.202	0.197	-0.151

Appendix L

Consent to Participate Form

This appendix provides the consent form for the experiment for the collection of the rule based dataset. The following pages reproduce the full consent form used.

CONSENT TO PARTICIPATE IN
NON-BIOMEDICAL RESEARCH

Predicting Procedural Training Effectiveness in Supervisory Control

You are asked to participate in a research study conducted by Professor Mary Cummings, Ph.D. from the Aeronautics and Astronautics Department at the Massachusetts Institute of Technology (M.I.T.). You were selected as a possible participant in this study because the expected population this research will influence is expected to contain men and women between the ages of 18 and 50 with an interest in using computers. You should read the information below, and ask questions about anything you do not understand, before deciding whether or not to participate.

- **PARTICIPATION AND WITHDRAWAL**

Your participation in this study is completely voluntary and you are free to choose whether to be in it or not. If you choose to be in this study, you may subsequently withdraw from it at any time without penalty or consequences of any kind. The investigator may withdraw you from this research if circumstances arise which warrant doing so.

- **PURPOSE OF THE STUDY**

The purpose of this study is to investigate procedural adherence in training and operational environments.

- **PROCEDURES**

If you volunteer to participate in this study, we would ask you to do the following things:

- Participate in a training period to learn the nuclear power plant control simulation interface and complete a 10 minute practice session to familiarize yourself with the power plant parameters.
- Participate in three 30-minute training sessions in which you will gain further understanding of the control of the power plant. You will work alongside two or three other participants to simulate a training course with multiple trainees, though you will each have your own workstations with your own nuclear power module to control.
- Participate in a 90-minute test session in which your performance will be evaluated to assess the learning that has taken place in the training sessions. You will need to use some of the skills and information learned in training for this test session.
- You will be awarded a score for the trial based on your accuracy in following procedures in the test session.
- All testing will take place at MIT in room 35-220.
- Total expected time: 4.5 hours

- **POTENTIAL RISKS AND DISCOMFORTS**

There are no anticipated physical or psychological risks involved in this study.

- **POTENTIAL BENEFITS**

While you will not benefit directly from this study, the results from this study will assist in the evaluation of training for supervisory control systems.

- **PAYMENT FOR PARTICIPATION**

You will be paid $25/hr to participate in this study, which will be paid upon completion of your debrief. Should you elect to withdraw in the middle of the study, you will be compensated for the hours you spent in the study. An additional $200 gift certificate to Best Buy will be awarded to the participant with the highest performance score.

- **CONFIDENTIALITY**

Any information that is obtained in connection with this study and that can be identified with you will remain confidential and will be disclosed only with your permission or as required by law. You will be assigned a subject number that will be used on all related documents to include databases, summaries of results, etc.

- **IDENTIFICATION OF INVESTIGATORS**

If you have any questions or concerns about the research, please feel free to contact the Principal Investigator, Mary L. Cummings, through phone: (617) 252-1512, e-mail: missyc@mit.edu, or mailing address: 77 Massachusetts Avenue, Room 33-311, Cambridge, MA, 02139. The investigators are Alexander Stimpson and Hosea Siu. They may be contacted at (352) 256-7455 or via email at ajstimps@mit.edu and hoseasiu@mit.edu respectively.

- **EMERGENCY CARE AND COMPENSATION FOR INJURY**

If you feel you have suffered an injury, which may include emotional trauma, as a result of participating in this study, please contact the person in charge of the study as soon as possible.

In the event you suffer such an injury, M.I.T. may provide itself, or arrange for the provision of, emergency transport or medical treatment, including emergency treatment and follow-up care, as

needed, or reimbursement for such medical services. M.I.T. does not provide any other form of compensation for injury. In any case, neither the offer to provide medical assistance, nor the actual provision of medical services shall be considered an admission of fault or acceptance of liability. Questions regarding this policy may be directed to MIT's Insurance Office, (617) 253-2823. Your insurance carrier may be billed for the cost of emergency transport or medical treatment, if such services are determined not to be directly related to your participation in this study.

- **RIGHTS OF RESEARCH SUBJECTS**

You are not waiving any legal claims, rights or remedies because of your participation in this research study. If you feel you have been treated unfairly, or you have questions regarding your rights as a research subject, you may contact the Chairman of the Committee on the Use of Humans as Experimental Subjects, M.I.T., Room E25-143B, 77 Massachusetts Ave, Cambridge, MA 02139, phone 1-617-253 6787.

SIGNATURE OF RESEARCH SUBJECT OR LEGAL REPRESENTATIVE

I understand the procedures described above. My questions have been answered to my satisfaction, and I agree to participate in this study. I have been given a copy of this form.

_____
Name of Subject

_____
Name of Legal Representative (if applicable)

_____ _____
Signature of Subject or Legal Representative Date

SIGNATURE OF INVESTIGATOR

In my judgment the subject is voluntarily and knowingly giving informed consent and possesses the legal capacity to give informed consent to participate in this research study.

_____ _____
Signature of Investigator Date

Appendix M

Demographic Survey

This appendix provides the demographic survey for the experiment for the collection of the rule based dataset. The following pages reproduce the survey form used.

DEMOGRAPHIC SURVEY

1. Subject number:_____
2. Age:_____
3. Gender: *M* *F*
4. Color Blindness: *N* *Y* If yes, type:_____
5. Occupation:_____

   if student, (circle one): *Undergrad* *Masters* *PhD*

   expected year of graduation:_____
6. Nuclear or conventional power plant experience (circle one): *No* *Yes*
   If yes, which plant:_____

   Level of Training _____ Licensed? *Y* *N* Years of experience:_____
7. Have you used detailed procedures before (e.g. checklists, model-building)?

   No *Yes*

   If yes, please briefly explain:_____

   _____
8. How often do you play computer games?

   Rarely *Monthly* *Weekly* *A few times a week* *Daily*

   Types of games played:_____
9. Rate your comfort level with using computer programs.

   Not comfortable *Somewhat comfortable* *Comfortable* *Very Comfortable*
10. What is your perception toward nuclear power plants?

    Intense dislike *Dislike* *Neutral* *Like* *Really Like*

Appendix N

Powerpoint Tutorial Slides

This appendix provides the powerpoint slides for the experiment for the collection of the rule based dataset.

Nuclear Power Plant Control Simulator Familiarization

Overview

These slides will familiarize you with the interface and procedures you will be using over the next several hours. If you have questions at any time during the familiarization session, please ask the experimenter.

- Ask questions if you have them
- Take your time

Primary Task

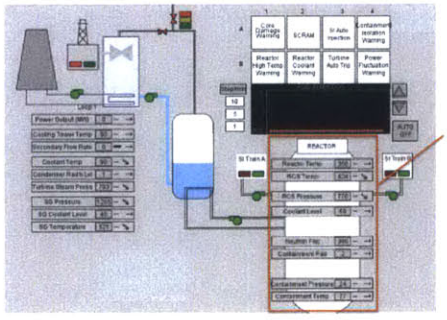
The experiment is divided into two sections:

1) Operator training (three 30-min modules)
   - In this section you will be shown how to perform specific tasks as a reactor operator
   - You then will have a chance to perform those tasks on your own (though you may ask questions)
   - After each module you will take a brief quiz to check your understanding of the procedures (including details in this presentation)
2) A test module (90 min) to assess what you have learned in training
   - In this section your objective is to monitor the reactor and appropriately deal with any issues that may arise by correctly following the appropriate procedures

Reactor

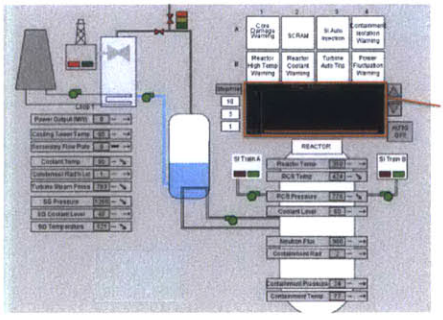
- For both training and test modules, you control your own simplified reactor
  - Participants in the room will work independently, and your actions will not affect any reactor other than your own
- Your score will be based in part on how well you follow the procedures given to you
  - Try to follow these procedures as closely as possible for each module
- First we will go over the basic operation of the nuclear reactor

Basics of a Nuclear Reactor



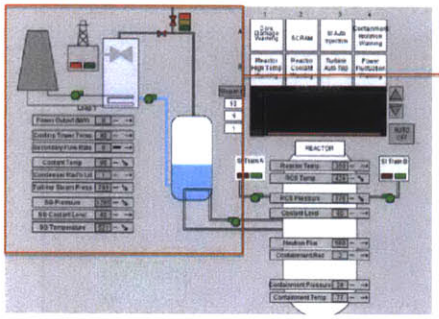
This is the reactor. Nuclear fission within the reactor produces heat that is used to generate electricity.

Basics of a Nuclear Reactor



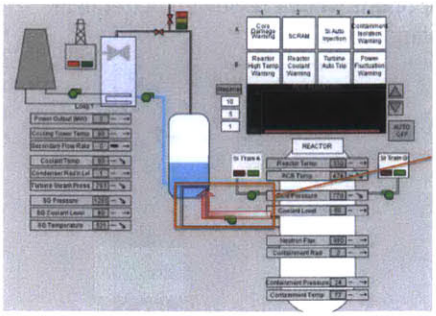
These are the control rods, which can slow or stop the reaction. When fully extracted (at level 0), the reaction and heat production is at maximum. When fully inserted (at level 100), the reaction halts and very little heat is produced.

Basics of a Nuclear Reactor



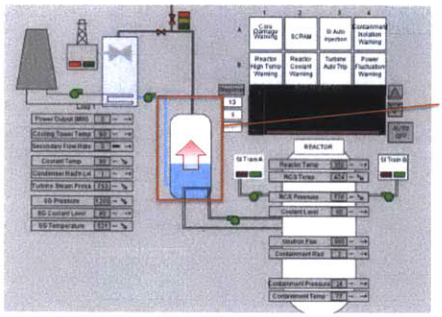
This is a Steam Generator Loop. It produces power from heat generated in the reactor. Note that there are 4 independent loops for the single reactor.

Basics of a Nuclear Reactor



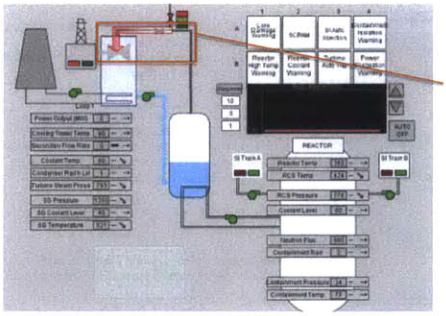
The fission reaction in the core heats water that is pumped through the Reactor Coolant System (RCS).

Basics of a Nuclear Reactor



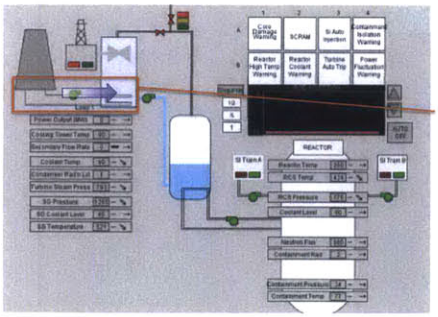
The hot water from the RCS boils water in the Steam Generator (SG)

Basics of a Nuclear Reactor



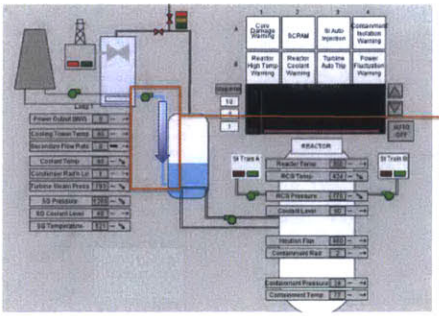
The steam from the SG flows through the steamline valve into the turbine, which produces power.

Basics of a Nuclear Reactor



Cool water pumped from the cooling tower condenses the steam coming out of the turbine.

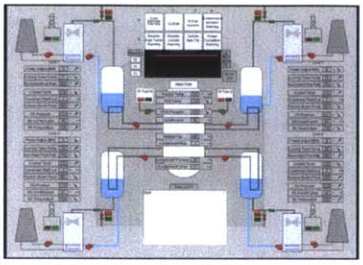
Basics of a Nuclear Reactor



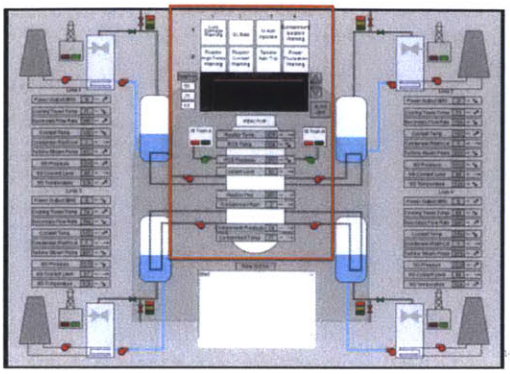
The condensed water returns to the Steam Generator to be boiled again.

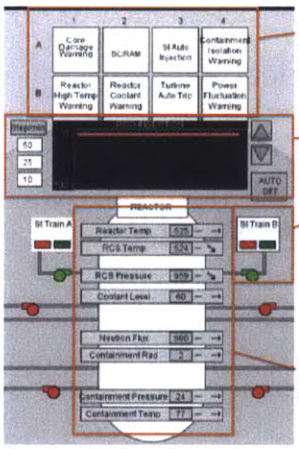
Main Interface

- Each component will be reviewed in detail in the following slides.



The middle of the interface represents the reactor, which is central to the nuclear power plant operation. We will now review this section of the interface.





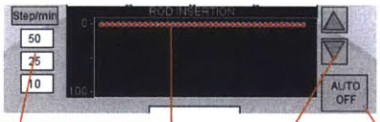
Annunciator Panel: Shows the warnings present in the nuclear power plant. When active, the tile will turn red. Each annunciator is identified by a letter-number combination (e.g. A3).

Control Rods: Control rods control the nuclear reaction in the reactor. They can be adjusted either automatically using the Step/min options or manually by clicking the up/down arrow.

Safety Injection (SI) trains: SI trains are an important safety feature. The status lights indicate whether the SI train is active (green) or in standby (red). Both the SI trains and the SI train pumps need to be active for the SI system to cool the reactor.

Reactor gauges: The gauges over the reactor are important measurements that should be monitored for safety.

Control Rod Controls



The step buttons indicate the rate at which the rods will be automatically extracted/inserted in steps per minute. This function is only active if the AUTO ON button is active. There are 100 steps between fully extracted and fully inserted.

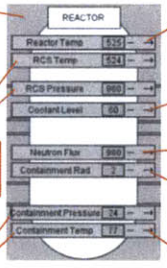
The Auto button turns the automatic retraction of rods on or off.

The arrows move ALL the rods in or out to change the reaction.

The rod positions are indicated by a bank of lights. All rods move as one unit. This position directly influences the nuclear reaction in the reactor.

Reactor Gauges

The reactor has a variety of measurements that are taken once every second. It is important to monitor these measurements.



Reactor Temperature is an important parameter, since uncontrolled increases could result in a meltdown.

Too low reactor coolant level will cause the reactor temperature to rise.

Neutron flux is related to the power output of the reactor.

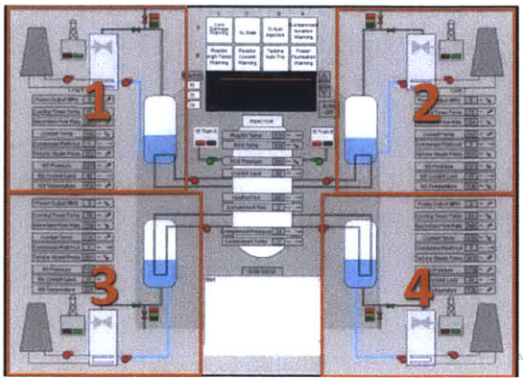
It is important to ensure radiation levels do not become unstable.

The arrows in the gauges represent the trend of the measurement. These arrows indicate the measurements are steady.

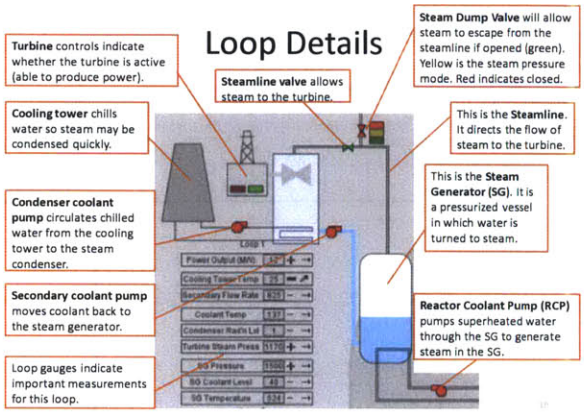
RCS is the Reactor Coolant System. This system provides cooling to the reactor core.

The containment area encapsulates the reactor and RCS systems.

The interface has 4 loops that operate in the same way. Each loop depends upon the reactor but operates independently. We will review the features of one loop in the next slides.

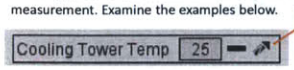


Loop Details

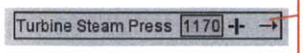


Gauge Details

Gauges have 4 components, from left to right: the gauge name, the measurement (in the box), whether the measurement is within limits or out, and the trend of the measurement. Examine the examples below.



This gauge indicates that the Cooling Tower Temperature is **below** limits (bold - sign) but is increasing over time (up arrow).



This gauge indicates that the Turbine Steam Pressure is **above** limits (bold + sign) and is steady at that pressure (horizontal arrow).

This gauge indicates that the Coolant Temperature is **within** limits and is steady at that temperature (horizontal arrow).

A downward trend is also possible, indicated by an arrow pointing down and to the right.
NOTE: The limits for each of the gauges on the interface are given in Appendix A of the Nuclear Reactor Operating Procedures contained in the white binder

Controls

- The main reactor controls are:
  - Pumps
  - Valves
  - Control rod controls (covered previously)
  - SI trains
  - Turbines
- For pumps and valves, simply click the pump or valve to change its status.
- For SI trains and Turbines, click the associated box with the lights on it to change its status

Pumps and Valves

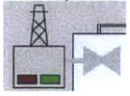
- Below is a pump. **Red means active (on). Green means inactive (off)**- note that this is the only exception to the normal red/green off/on scheme.

- Below is a valve. **Red means closed. Green means open.**

SI Trains and Turbines

- Below is an SI Train. **Red means inactive (off). Green means active (on).**

- Below is a Turbine. **Red means inactive (off). Green means active (on).** When inactive, the loop will produce no power.



SCRAM

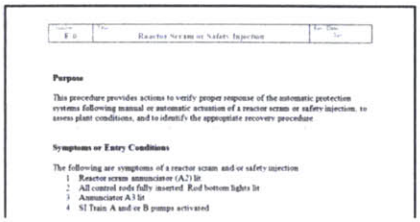
- A SCRAM refers to an emergency shutdown of the reactor
- When a major problem is detected, the reactor may automatically fully insert the control rods (to level 100) to stop the reaction – this is a SCRAM
- There are several possible causes for a SCRAM, and these are covered in training module 3

Reactor Procedures

- Please look at the white Procedures binder located at your workstation.
- Procedures are carefully written to guide you step-by-step through situations you may encounter.
- Follow them closely to complete the training modules and fix any problems with the reactor, should they arise
- Remember that part of your score will be based on how well you follow the procedures
- Each procedure has entry conditions for use, which will be discussed next

Entry Conditions

- Each procedure in the manual has a set of entry conditions
- You should **only use a procedure set if you have met the entry conditions**
- Entry conditions involve information gathering only, no interaction with the interface is necessary
- Once entry conditions have been met, turn the page to start the procedure



E-0 Reactor Scram or Safety Injection

Purpose

This procedure provides actions to verify proper response of the automatic protection systems following manual or automatic actuation of a reactor scram or safety injection, to assess plant conditions, and to identify the appropriate recovery procedure.

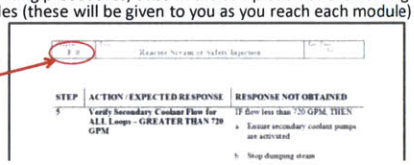
Symptoms or Entry Conditions

The following are symptoms of a reactor scram and/or safety injection
1 Reactor scram annunciator (A2) lit
2 All control rods fully inserted. Rod bottom lights lit
3 Annunciator A3 lit
4 SI Train A and/or B pumps activated

Types of Procedures

- The nature of the procedures can be identified by the category in the upper left
  - E = emergency procedures, only to be used if there is a reactor SCRAM
  - M = maintenance procedures, used for contacting and coordinating with maintenance crews
  - AOP = Anomalous Operating Procedures, to be used if the system is exhibiting unusual behavior but has not SCRAMed
  - T = Training procedures, used in the completion of the training modules (these will be given to you as you reach each module)

Indicates that this is an emergency procedure

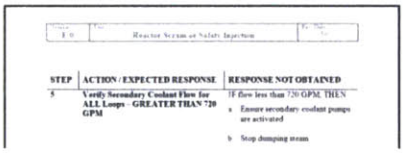


E-0 Reactor Scram or Safety Injection

STEP	ACTION / EXPECTED RESPONSE	RESPONSE NOT OBTAINED
5	**Verify Secondary Coolant Flow for ALL Loops – GREATER THAN 720 GPM**	IF flow less than 720 GPM, THEN a Ensure secondary coolant pumps are activated b Stop dumping steam

Action/Expected Response

- Procedure steps have 2 parts:
  - Action / Expected Response
  - Response not obtained
- Follow the step in the left column (Action / Expected Response).
- If those conditions or actions are not true, go to the right hand column (response not obtained)to complete that step.
- Continue with the next step in the Action column.



E-0 Reactor Scram or Safety Injection

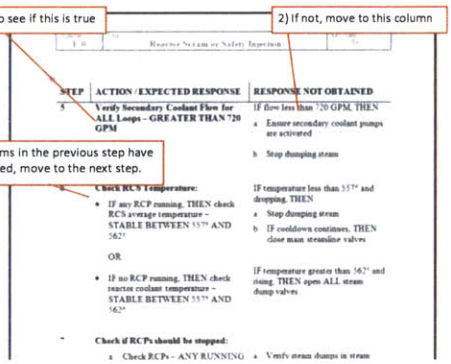
STEP	ACTION / EXPECTED RESPONSE	RESPONSE NOT OBTAINED
5	**Verify Secondary Coolant Flow for ALL Loops – GREATER THAN 720 GPM**	IF flow less than 720 GPM, THEN a Ensure secondary coolant pumps are activated b Stop dumping steam

Action/Expected Response Example

1) Check to see if this is true

2) If not, move to this column

3) After all items in the previous step have been completed, move to the next step.



E-0 Reactor Scram or Safety Injection

STEP	ACTION / EXPECTED RESPONSE	RESPONSE NOT OBTAINED
5	**Verify Secondary Coolant Flow for ALL Loops – GREATER THAN 720 GPM**	IF flow less than 720 GPM, THEN a Ensure secondary coolant pumps are activated b Stop dumping steam
	Check RCS Temperature: • IF any RCP running, THEN check RCS average temperature – STABLE BETWEEN 557° AND 562° OR • IF no RCP running, THEN check reactor coolant temperature – STABLE BETWEEN 557° AND 562°	IF temperature less than 557° and dropping, THEN a Stop dumping steam b IF cooldown continues, THEN close main steamline valves IF temperature greater than 562° and rising, THEN open ALL steam dump valves
	Check if RCPs should be stopped: a Check RCPs – ANY RUNNING	a Verify steam dumps in steam

Response Not Obtained

- Move to the "response not obtained" column as soon as any step or sub-step within the "action/expected response" column is not true
- Once you have completed the "response not obtained" procedure, move on to the "action/expected response" column for the next step… do not return to the "action/expected response" column for the same step

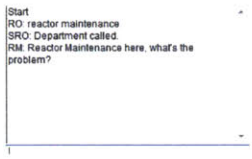
Verification

- Several of the procedures will ask you to verify states of the loops or the rods
- In order to complete the verification for that step, please right click on the item to be verified
- You will hear a tone indicating that the item has been verified, and meters will flash black
- Example: "Verify RCS Temp"

Verification

- Entry conditions also ask for you to check states of the interface
- However, do NOT click the interface when checking entry conditions (checking is NOT the same as verifying)
- Even if the object you are verifying is not in the expected state, be sure to right click on it to indicate that you have completed the verification – then move to the "response not obtained" column

Chat Box

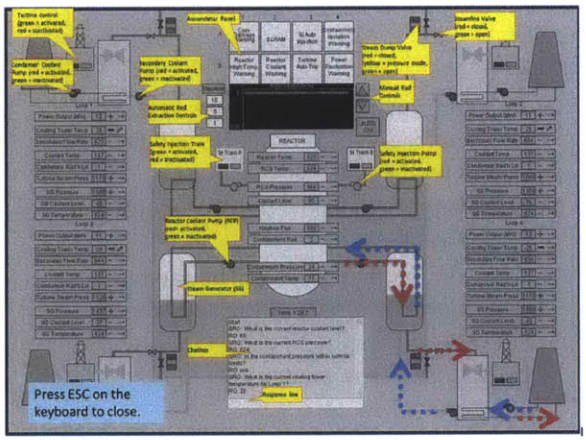


The chat box allows you to communicate with the supervisor and the maintenance departments. Note that standard procedures for contacting maintenance are contained in procedure **M-1** in your binder.

- RO = Reactor Operator (you)
- SRO = Senior Reactor Operator (supervisor)
- RM = Reactor Maintenance

You may be required to communicate using the chat box. Monitor the box throughout the test module, and respond as clearly and concisely as possible when appropriate. Type any needed messages into the white box and press ENTER. There are other departments besides RM that you may have to contact. A listing of relevant departments is provided in Appendix B of your operating procedures binder.

Typing in "help" to the chat box will pull up a labeled diagram of the reactor interface for your reference. Exit this interface with the *ESC* key. A sample screenshot of the reference interface is shown on the next slide.



Scoring

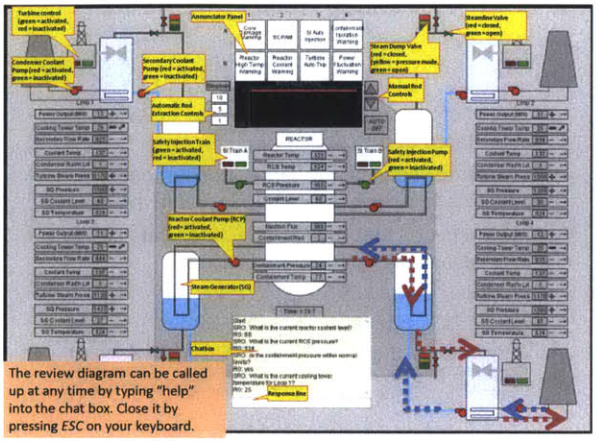
- Your score will be based two elements:
  - How accurately you follow the procedures given
  - Average power output (for the final test module)
- Any deviations from the given procedures will result in point deductions
- You will not be penalized for using the reference screen by typing "help" in the chatbox

Training Modules

- Each of the 3 modules are divided up into five parts
  - Training Video – provides information about the module's topic
  - Practice Session – uses the procedures that will be given to you for each module
  - Post-session survey – asks you to rate some of the aspects of the module
  - Multiple choice test – assesses your learning for that module
  - Review slides – reviews the material on the test
- If you are finished with one part of the module, let the training supervisor know so he can start you on the next part or next module

Review

- The next slide summarizes all the components in the interface.



The review diagram can be called up at any time by typing "help" into the chat box. Close it by pressing *ESC* on your keyboard.

Summary

- Your performance will be assessed by how well you follow the procedures and the power output on the test module
- Therefore, try to follow the procedures as closely as possible
- Your performance score will be used to determine the winner of the $200 Best Buy gift card
- Do <u>NOT</u> close any windows.

Appendix O

Post-Module Quizzes

This appendix provides the post-module quizzes in the experiment for the collection of the rule based dataset. The following pages reproduce the survey form used.

Module 1 Test Questions

INSTRUCTIONS: Circle the BEST answer for each question.

QUESTION 1: If instructed to insert control rods into the reactor and automatic insertion is not functional, which of the following steps would be your first course of action?

a) attempt to manually insert the control rods
b) enter a maintenance module
c) leave control rods where they are and proceed with normal operations
d) attempt to automatically extract rods completely, then try automatic insertion again

QUESTION 2: Which one of the following is not directly associated with power output reduction?

a) decreasing reactor coolant system pressure
b) turbines deactivated
c) rod insertion
d) rising SG pressure

QUESTION 3: Which of these transports hot water from the reactor to the steam generators?

a) Secondary Coolant Pumps (SCPs)
b) Reactor Coolant Pumps (RCPs)
c) turbines
d) steam dump valves

QUESTION 4: Which color indicates that a pump is activated?

a) red
b) green

QUESTION 5: Which color indicates that a valve is open?

a) red
b) green

QUESTION 6: Why do SG steamline valves have to be open to initiate power production?

a) they allow radiation from the reactor to enter the cooling towers
b) they allow steam from the steam generator tanks to pass through the turbines
c) they allow the reactor to be turned on
d) they allow reactor coolant levels to rise

Module 2 Test Questions

INSTRUCTIONS: Circle the BEST answer for each question.

QUESTION 1: What is an expected response to deactivating an RCP?

a) reactor temperature would decrease
b) condenser radiation levels would increase
c) SG pressure for that loop would decrease
d) Turbine Auto Trip indicator would turn red

QUESTION 2: What is the main purpose of the Safety Injection system?

a) to cool down the reactor core in case of extreme overheating
b) to reinforce the containment building in case of a radiation leak
c) to decrease power generation
d) to increase reaction rate for higher reactor temperatures

QUESTION 3: Which of the following alarms should result from the activation of SI trains and pumps?

a) Core Damage Warning
b) Turbine Auto Trip
c) SCRAM
d) SI Auto Injection

QUESTION 4: What is an expected response to deactivating a loop's secondary coolant pump?

a) rising SG temperature
b) rising condenser radiation levels
c) SCRAM alarm
d) increased power output fluctuations

QUESTION 5: What is an expected response to a running loop if one were to open a steam dump valve?

a) power output for that loop would rise
b) power output for that loop would drop
c) condenser radiation levels would drop
d) containment radiation levels would drop

Module 3 Test Questions

INSTRUCTIONS: Circle the BEST answer for each question or fill in the correct response.

QUESTION 1: What was the issue that caused the SCRAM? _____

QUESTION 2: Which of the following conditions must be met for Module 3 to be used?

a) Reactor Scram Annunciator is lit and the containment pressure is rising
b) control rods have been fully inserted and the Reactor Scram Annunciator is lit
c) control rods have been fully inserted and the Reactor Scram Annunciator is off
d) reactor temperature dropping and the Core Damage Warning is lit

QUESTION 3: Once a scram has been verified, what state should the turbine indicators be in?

a) turbine trip (red light)
b) turbine trip (green light)

QUESTION 4: Which of the following should be done if Secondary Coolant Flow for all loops is less than 720 GPM after a reactor scram?

a) call for maintenance on the most affected loop
b) deactivate turbines
c) open any closed steam dump valves
d) close any open steam dump valves

QUESTION 5: If condenser radiation levels are not normal, which of the following is likely to be responsible?

a) faulted steam generator
b) steam generator tube rupture
c) loss of reactor
d) loss of secondary coolant

QUESTION 6: Which indicator color shows that SI Trains have been reset?

a) green
b) red

Appendix P

Post-Experiment Questionnaire

This appendix provides the post-experiment questionnaire used in the experiment for the collection of the rule based dataset..

POST EXPERIMENT SURVEY

1. How confident were you about the actions you took?

Not Confident *Somewhat Confident* *Confident* *Very Confident* *Extremely Confident*

Comments:

2. How would you rate your performance?

Very Poor *Poor* *Satisfactory* *Good* *Excellent*

3. How stressed did you feel during the alarm situation?

Not Stressed *Somewhat Stressed* *Stressed* *Very Stressed* *Extremely Stressed*

4. How busy did you feel during the alarm situation?

Idle *Not Busy* *Busy* *Very Busy* *Extremely Busy*

5. Do you feel that the training sufficiently prepared you for the test? *No* *Yes*

Comments:

6. How well do you feel you understand nuclear power plant operation?

Very Poorly *Poorly* *Satisfactory* *Well* *Very Well*

7. Were the procedures easy to understand? *No* *Yes*

Comments:

8. Other comments:

Appendix Q

Measuring Procedure Adherence and Development of the Procedure Adherence Metric (PAM)

Q.1 Measuring Procedure Adherence

Procedure adherence represents the faithfulness to which the operator follows the prescribed procedures. Prior research on adherence to procedures has primarily been part of studies on human error in rule-based environments [155–157]. This research has focused on the theoretical aspects of errors while following procedures and the modeling of human error in these cases, such as the Memory For Goals (MFG) model [158, 159]. This work identifies differences in tendencies for adherence across cultures and levels of experience, and often utilizes interviews with operators rather than experimental studies (e.g. [155, 160]). A few experimental studies have been conducted, primarily focusing on the nature of errors in procedure-following or improvement of adherence through the design of procedures or new interfaces [161, 162]. There are few sources that discuss the measurement of procedure adherence, and these do not discuss the implications of procedure adherence as an assessment metric in training. Thus, one of the important contributions of this work includes the discussion of potential measures of procedure adherence, and their use both as a training assessment metric and as features in machine learning approaches. These topics are covered in this and the following sections.

Adherence to procedures can have several interpretations dependent upon the partic-

ular domain in which the procedures are used. In nuclear power plant operation, there are often many procedures available contained in either binders or more recently in computerized checklist systems. Thus, an important aspect of adherence in nuclear power plants is the ability to select the correct procedure to be used based on the current system state. This ability is referred to in this appendix as "procedure selection." When an incorrect procedure is selected, subsequent actions performed in the following of that procedure are unlikely to be appropriate for the situation.

Once a procedure is selected, a second type of adherence (simply termed "adherence" here) arises relating to the accurate completion of all appropriate steps contained in that procedure. Sometimes procedures will be branched, in which case not all steps will be used. In these cases, adherence can be measured to following all the steps along one set of sequential actions, i.e. the "path" through the procedure. Sometimes only a single path is correct given the system state; other times there may be multiple "correct" paths for appropriate operation. In the former case, the procedure is linear, and at any time during the course of completing the procedure there is only one intended action for the next step in the procedure. Thus, adherence can be measured based on whether the next action is the intended action. When multiple correct paths exist, adherence is much more difficult to measure, as actions that are incorrect for one path may still be correct along a different path.

There are a variety of metrics that could be used to measure procedure selection and adherence. For procedure selection, the primary information of interest is whether or not the correct procedure was selected. The simplest assessment metric in training could then be a binary of whether the trainee did ("1") or did not ("0") choose the correct procedure. Another option would be to apply a performance penalty to those trainees who did not select the correct procedure, which would typically manifest itself as a weighted case of the binary metric (e.g. "-10 points if they did not select the correct procedure"). In machine learning, any transformation used (such as range or z-score transformations described in Chapter 3) automatically reweights features to an approximately equal scale and would thus account for any weighting applied to the binary metric. Thus, procedure selection in this analysis is assessed by the simple binary metric rather than any weighted form.

In the consideration of adherence, it is useful to think about a procedure as a sequence, defined as an "ordered set of events." An SOP defines a series of actions for the

user to take, typically under a certain set of initial conditions that make the procedure appropriate. The set of actions contained in a procedure can be translated into a sequence, with each action having a previous action and a subsequent action (see Figure Q-1). A trainee similarly generates an ordered sequence of actions as s/he attempts to complete the procedure as part of the training module. In a sequence of actions, common errors include omission of an action, performing actions out of order, or substitution of an action with an incorrect one. These errors create mismatches between the procedure sequence and the trainee sequence. Typically in a complex system, there are more actions available to the user than are needed for any particular procedure. Consider the aircraft training example given in Chapter 3; during the pre-takeoff checklist the pilot is not required to press every button in the cockpit. With more actions available than used in the procedure, it is possible for a trainee sequence to contain actions that are never observed in the procedure sequence.

In this framework, procedure adherence can be measured by the difference between a

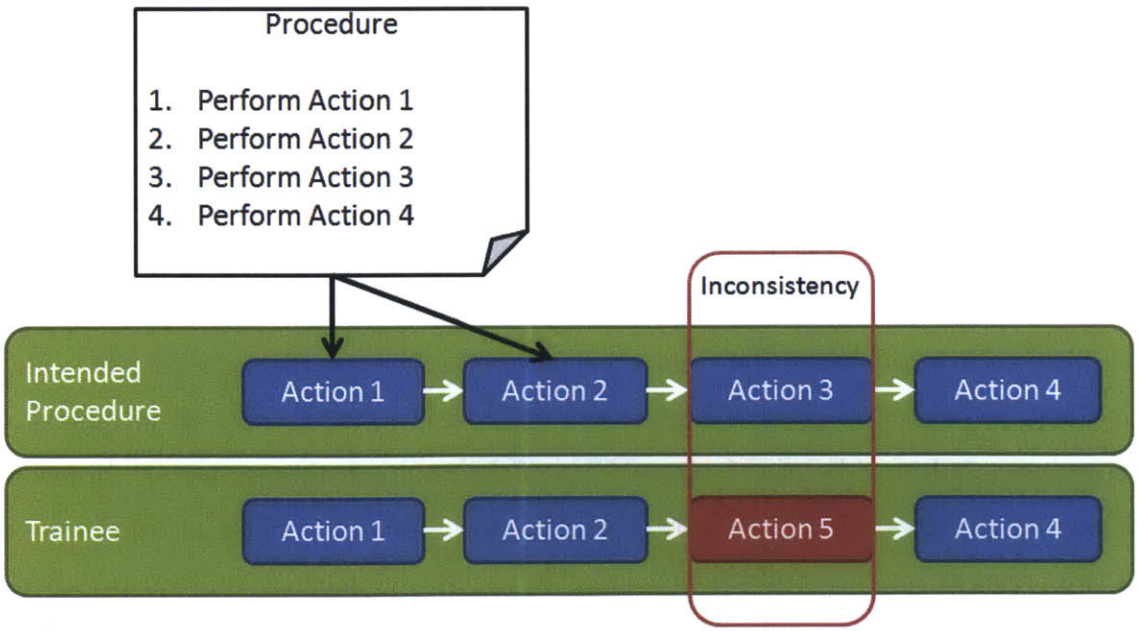


Figure Q-1: Viewing a procedure as a sequence.

SOP sequence and the trainee sequence. Numerous methods that measure the distance between sequences have been developed, including sequence-based methods (e.g. Levenshtein distance [163]) and model-based methods (e.g. Kullback-Leibler divergence [164]). Sequence-based methods focus on the direct comparison of the sequences, while model-based methods model each sequence and then compare the similarity of the models as a proxy for sequence distance. To select the best method for the calculation of sequence distance, four elements important to procedure adherence measurement are:

- Ability for different types of errors to be penalized separately (i.e. error weighting)
- Non-equal sequence length between the observed and expected sequence of actions, i.e., where the user performs a different number of actions than the prescribed procedure
- Sensitivity to number of deviations between the observed versus expected sequence of actions
- Sensitivity to trainee action order: the ability to distinguish between correct and incorrect ordering

Three methods that calculate sequence distance were considered as possible features for machine learning approaches to measuring procedure adherence: two sequence-based methods (Levenshtein distance and suffix arrays [165]) and one model-based method (Kullback-Leibler divergence). The details of the calculation of each method and their comparison on the four criteria presented above can be found in Appendix C. Briefly, the Levenshtein distance is calculated by the minimum number of edits (insertions, deletions, or substitutions) to change the trainee sequence into the intended sequence. It is able to handle sequences of unequal lengths and is sensitive to the number of deviations of the trainee, and provides easy interpretation of the resultant distance values. However, it weights all errors equally and only considers action order at the level of transposition of two adjacent actions. Suffix arrays are able to identify matching subsequences beteween the trainee and prescribed sequence, and thus account for the ordering of actions. However, the distance measure focuses on the largest matching subsequence, and thus does not account directly for the number of deviations. The Kullback-Leibler (KL) approach exhibits similar adherence detection properties to the Levenshtein distance, but

additionally accounts for action ordering and weights errors based on the frequencies of actions and transitions encountered in the prescribed sequence. Thus, the KL divergence is able to meet all four criteria specified above.

Two metrics stand out as potential candidates for measuring adherence in training: the Levenshtein distance for its simplicity and ease of interpretation, and the KL divergence by meeting all of the desired criteria. However, the KL divergence requires additional modifications before it can be used directly for measuring procedure adherence. Thus, a new metric was created, termed the Procedure Adherence Metric (PAM), that captures the benefits of using the KL divergence approach but is able to be calculated directly from sequences generated by trainees in rule-based environments. The following section details the calcuation of the PAM.

Q.1.1 Procedure Adherence Metric (PAM)

The main goal of measuring procedure adherence is to assess trainees' performance against the SOP. Additionally, trainees can be objectively compared against each other based on their training performance, and tracking procedure adherence can indicate struggling trainees that need re-training. The Procedure Adherence Metric (PAM) was based on the KL divergence between the trainee and intended action sequences. Formally, the KL divergence between two sequences can be calculated as shown in Equation Q.1.

$$D_{KL}(P||Q) = \sum_i ln(\frac{P(i)}{Q(i)})P(i) \tag{Q.1}$$

where $P(i)$ is the probability of observing the transition from action $i-1$ to action i in one sequence (such as the intended sequence), and $Q(i)$ is the probability of observing the transition from action $i-1$ to action i in the other sequence (such as the trainee sequence). As can be seen from this formula, the KL divergence requires at least two actions in the sequence, as no transitions exist with sequences containing only one action. Also, the basic form of the KL divergence is not symmetric, meaning that $D_{KL}(P||Q)$ is not necessarily equivalent to $D_{KL}(Q||P)$. For calculating adherence, it is useful to have a metric that does not depend upon which sequence is used for P and Q, and thus a symmetrized form can be used, as shown in Equation Q.2.

$$D_{symmetricKL} = D_{KL}(P||Q) + D_{KL}(Q||P) \tag{Q.2}$$

In this symmetrized form, the divergence will be the same regardless of which sequence is used for P and Q. An adherence metric based on this divergence could use a single final value calculated after the trainee has completed the module, or could utilize KL divergence partway through the the training module. Consider a trainee who completes 30 actions during a training module. At any given point during the training module, the "current" KL divergence between the trainee sequence and the intended sequence at that point in the module can be calculated. Thus a sequence of KL divergences can be generated over time, but it is unclear which aspects of this divergence sequence are the most useful for measuring adherence.

KL divergence can be measured in many ways: maximum KL divergence, the mean KL divergence, the final KL divergence (using only the full sequence), and the sum of the KL divergences in the sequence. It was determined that using the sum of the KL divergence values over all actions in a training module provided the closest rankings of trainees (from best to worst) as compared to an expert evaluator (Appendix D). Using the sum of KL divergences as the basis for the adherence metric, the PAM value can then be calculated as shown in Equation Q.3.

$$PAM = \sum_{i=1}^{N} D_{KL} \tag{Q.3}$$

where N is the number of events or actions in the training sequence, and D_{KL} represents the symmetrized Kullback-Leibler divergence between the trainee sequence of states $1i$ and the intended sequence of states of equivalent length. If N is greater than the number of states in the intended sequence (M), the complete intended sequence is used for all $i > M$. It is important to recognize that as the PAM is based on divergence, and a lower score indicates better performance.

A potential issue that arises in the use of the KL divergence for the PAM is zero-probability values in the transition matrix. This transition matrix represents the probability of all transitions between states in the model. For a model based on a sequence, the maximum likelihood estimate simply counts the number of times a consecutive action pair is found (e.g. action 1 to action 2), and normalizes by the number of transitions.

An example based on three possible actions is shown in Figure Q-2. If a particular set of actions are never observed consecutively (such as action 1 to action 3 in Figure Q-2), the count (and therefore the probability estimate) for that transition is zero. The size of the state transition matrix is heavily dependent on the number of existing actions (NxN for N actions), and can be large for CBT settings. Even with only three actions in Figure Q-2, it would take a sequence of at least length 10 (9 transitions) to have no zero-probability transitions.

Often the set of actual transitions in any particular training procedure will not cover the entire set of possible transitions. When included in the model, these zero probability events send the KL divergence to infinity. Instead a small (but non-zero) probability can be assigned to transitions that do not occur in the intended procedure. This results in a large divergence score (poor performance) in the PAM but does not send the divergence to infinity. Frequency estimation is a set of statistical techniques that provide estimates

Sequence: 1,1,2,2,3,3

Transition Counts

Starting State \ Ending State	1	2	3
1	1	1	0
2	0	1	1
3	0	0	1

Transition Probabilities

Starting State \ Ending State	1	2	3
1	0.5	0.5	0
2	0	0.5	0.5
3	0	0	1

Figure Q-2: Example action sequence with transition counts and probabilities. In the sequence, the numbers represent individual actions performed by the trainee.

of the probabilities of unobserved events, such as the zero-probability events in the transition matrix. Some of the most common frequency estimation methods include additive smoothing and Good-Turing estimation. Additive smoothing simply adds new transition counts such that there are no zero-probability transitions. This strategy works well when there are only a few unobserved events, but can dramatically alter the overall distribution if there are many zero-probability events such as is observed in CBT data. Good-Turing estimation estimates the probability of novel events based on the number of infrequently observed events, and thus self-corrects for cases where the transition matrix is sparse (few non-zero values). Thus for the PAM, Good-Turing smoothing was selected based on its ability to handle large numbers of novel events.

Q.1.2 Adherence Metrics as Features

Of the wide range of adherence metrics that can be used as features in machine learning, the Levenshtein distance was selected for its simplicity in interpretation, and the PAM was selected for its ability to meet the four criteria stated earlier in this appendix. PAM is calculated as described above, while Levenshtein distance is calculated by the minimum number of edits (insertions, deletions, or substitutions) to change the trainee sequence into the intended sequence. For each of these metrics, the value after each action was calculated, representing the process-level features. The final value at the end of the module provided the summative-level features.

Adherence metrics such as the PAM or Levenshtein distance can be utilized as features both at a process-level (for each action) or at a summative-level (end of module). While calculating the metric at each action would provide a high number of features for use in machine learning methods, as discussed in Chapter 3, having too many features compared to data points may cause overfitting of the models. Thus, two approaches can be taken: using only summative-level metrics or a reduced set of process-level metrics based on feature selection techniques. In this analysis, both approaches were taken and the results are presented in Chapter 4.

To illustrate the calcuation of these values, consider an example where the prescribed sequence is AABBCC, and the trainee sequence is ABBCDC. In this example, the trainee has made several mistakes: they have omitted an "A" action and have included an extraneous "D" action that does not show up in the prescribed sequence. Both the

Levenshtein distance and PAM can be used as example adherence metrics to describe the timing and severity of the errors. Table Q.1 shows the progression of both the Levenshtein distance and PAM over the course of this sequence. Both Levenshtein and PAM increase as the trainee moves through the sequence and commits the errors. The Levenshtein distance essentially acts to "count" the number of errors made by the trainee. At action 2 where the trainee omits an "A" action and at action 5 where the trainee includes an additional "D" action, the Levenshtein distance increases by one, resulting in a final value of 2. The PAM shows additional fluctuations based on comparing the flow of the sequences as a whole, through the comparison of the transitions between actions. By accounting for transitions (and thus action ordering), there is a relatively small penalty for the PAM at action 4, where the subsequence ABBC is seen in both the trainee and intended sequences. A much harsher penalty is given by the PAM to the added "D" action, as it results in transitions both to and from the "C" action which are unseen in the intended sequence. Both are valid representations of the adherence of the trainee through time, and thus were included as potential features in machine learning approaches. Note that if this sequence comprised the full module, the resultant summative adherence scores would be 2 and 5.99 for Levenshtein distance and PAM, respectively. Both the summative and process-level (action by action) values for the adherence metrics are utilized as possible adherence features in the machine learning sections presented in Chapter 4.

Table Q.1: Progression of Levenshtein and PAM over example sequences

Action Number	1	2	3	4	5	6
Intended Action	A	A	B	B	C	C
Trainee Action	A	B	B	C	D	C
Levenshtein Distance	0	1	1	1	2	2
PAM	0	2.04	2.68	1.70	5.06	5.99

Bibliography

[1] J. J. Phillips and S. L. Oswald, *Recruiting, training, and retaining new employees: Managing the transition from college to work.* Jossey-Bass Publishers, 1987.

[2] J. M. Barron, J. Bishop, and W. C. Dunkelberg, "Employer search: the interviewing and hiring of new employees," *The Review of Economics and Statistics*, pp. 43–52, 1985.

[3] R. A. Noe, *Employee training and development.* McGraw-Hill/Irwin Boston, 2002.

[4] A. M. Rose, "Acquisition and retention of skills," in *Applications of human performance models to system design*, pp. 419–426, Springer, 1989.

[5] W. Form, "On the degradation of skills," *Annual Review of Sociology*, pp. 29–47, 1987.

[6] K. R. Murphy and J. Cleveland, *Understanding performance appraisal: Social, organizational, and goal-based perspectives.* Sage, 1995.

[7] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, "A model for types and levels of human interaction with automation," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 30, no. 3, pp. 286–297, 2000.

[8] D. Lawton, N. Vye, J. Bransford, E. Sanders, M. Richey, D. French, and R. Stephens, "Online learning based on essential concepts and formative assessment," *Journal of Engineering Education*, vol. 101, no. 2, pp. 244–287, 2012.

[9] I. E. Allen and J. Seaman, *Changing Course: Ten Years of Tracking Online Education in the United States.* ERIC, 2013.

[10] A. G. Picciano and J. Seaman, *K-12 Online Learning: A 2008 Follow-Up of the Survey of US School District Administrators.* Sloan Consortium, 2009.

[11] K. Kraiger, J. K. Ford, and E. Salas, "Application of cognitive, skill-based, and affective theories of learning outcomes to new methods of training evaluation," *Journal of Applied Psychology*, vol. 78, no. 2, p. 311, 1993.

[12] C. M. Bishop, *Pattern recognition and machine learning*, vol. 1. Springer New York, 2006.

[13] J. Rasmussen, "Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models," *Systems, Man and Cybernetics, IEEE Transactions on*, no. 3, pp. 257–266, 1983.

[14] "Training." `http://www.businessdictionary.com/definition/training.html`. Accessed: 2014-10-16.

[15] R. A. Bjork and E. L. Bjork, "Optimizing treatment and instruction: Implications of a new theory of disuse," *Memory and society: Psychological perspectives*, pp. 116–140, 2006.

[16] K. A. Ericsson, "The influence of experience and deliberate practice on the development of superior expert performance," in *Cambridge Handbook of Expertise and Expert Performance* (K. A. Ericsson, N. Charness, P. Feltovich, and R. Hoffman, eds.), pp. 685–706, Cambridge, UK: Cambridge University Press, 2006.

[17] R. R. Hoffman, "How can expertise be defined? implications of research from cognitive psychology," *Exploring expertise*, pp. 81–100, 1996.

[18] S. Kalyuga, P. Chandler, and J. Sweller, "Levels of expertise and instructional design," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 40, no. 1, pp. 1–17, 1998.

[19] S. Kalyuga and J. Sweller, "Rapid dynamic assessment of expertise to improve the efficiency of adaptive e-learning," *Educational Technology Research and Development*, vol. 53, no. 3, pp. 83–93, 2005.

[20] A. M. Lesgold, "Acquiring expertise," Tech. Rep. UPITT/LRDC/ONR/PDS-5, Pittsburgh University Learning Research and Development Center, 1983.

[21] J. R. Anderson, "Acquisition of cognitive skill," *Psychological Review*, vol. 89, no. 4, p. 369, 1982.

[22] P. M. Fitts and M. I. Posner, *Human performance.* Oxford, England: Brooks/Cole, 1967.

[23] D. Eseryel, "Approaches to evaluation of training: Theory & Practice," *Educational Technology & Society*, vol. 5, no. 2, 2002.

[24] K. Alvarez, E. Salas, and C. M. Garofano, "An integrated model of training evaluation and effectiveness," *Human Resource Development Review*, vol. 3, no. 4, pp. 385–416, 2004.

[25] P. N. Blanchard, J. W. Thacker, and S. A. Way, "Training evaluation: perspectives and evidence from Canada," *International Journal of Training and Development*, vol. 4, no. 4, pp. 295–304, 2000.

[26] R. E. Clark and A. Voogel, "Transfer of training principles for instructional design," *ECTJ*, vol. 33, no. 2, pp. 113–123, 1985.

[27] J. K. Ford and D. A. Weissbein, "Transfer of training: An updated review and analysis," *Performance Improvement Quarterly*, vol. 10, no. 2, pp. 22–41, 1997.

[28] D. Ghodsian, R. A. Bjork, and A. S. Benjamin, "Evaluating training during training: Obstacles and opportunities," *Training for a rapidly changing workplace: Applications of psychological research*, pp. 63–88, 1997.

[29] P. O'Connor, J. Campbell, J. Newon, J. Melton, E. Salas, and K. A. Wilson, "Crew resource management training effectiveness: a meta-analysis and some critical needs," *The International Journal of Aviation Psychology*, vol. 18, no. 4, pp. 353–368, 2008.

[30] R. Bjork and D. Druckman, *The untapped potential of training.* London, England: Chemical Industry Society, 1994.

[31] P. R. Sackett and E. J. Mullen, "Beyond formal experimental design: Towards an expanded view of the training evaluation process," *Personnel Psychology*, vol. 46, no. 3, pp. 613–627, 1993.

[32] E. Salas and J. A. Cannon-Bowers, "The science of training: A decade of progress," *Annual Review of Psychology*, vol. 52, no. 1, pp. 471–499, 2001.

[33] R. A. Schmidt and R. A. Bjork, "New conceptualizations of practice: Common principles in three paradigms suggest new concepts for training," *Psychological science*, vol. 3, no. 4, pp. 207–217, 1992.

[34] W. B. Whitten II and R. A. Bjork, "Learning from tests: Effects of spacing," *Journal of Verbal Learning and Verbal Behavior*, vol. 16, no. 4, pp. 465–478, 1977.

[35] F. Galton, *Hereditary genius.* Macmillan and Company, 1869.

[36] D. E. Rumelhart and D. A. Norman, "Analogical processes in learning," *Cognitive skills and their acquisition*, pp. 335–359, 1981.

[37] H. A. Simon and W. G. Chase, "Skill in chess," *American Scientist*, vol. 61, no. 4, 1973.

[38] R. R. Hoffman, *The psychology of expertise.* Springer New York, 1992.

[39] W. W. Lee and D. L. Owens, *Multimedia-based instructional design: computer-based training, web-based training, distance broadcast training, performance-based solutions.* John Wiley & Sons, 2004.

[40] N. Harrison, *How to design self-directed and distance learning: a guide for creators of web-based training, computer-based training, and self-study materials.* McGraw-Hill Companies, 1999.

[41] C.-L. C. Kulik and J. A. Kulik, "Effectiveness of computer-based instruction: An updated analysis," *Computers in human behavior*, vol. 7, no. 1, pp. 75–94, 1991.

[42] T. C. Williams and H. Zahed, "Computer-based training versus traditional lecture: Effect on learning and retention," *Journal of Business and Psychology*, vol. 11, no. 2, pp. 297–310, 1996.

[43] J. Orlansky and J. String, "Cost-effectiveness of computer-based instruction for military training," in *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*, vol. 1980, NTSA, 1980.

[44] A. N. Summers, G. C. Rinehart, D. Simpson, and P. N. Redlich, "Acquisition of surgical skills: a randomized trial of didactic, videotape, and computer-based training," *Surgery*, vol. 126, no. 2, pp. 330–336, 1999.

[45] S. S. Harrington and B. L. Walker, "A comparison of computer-based and instructor-led training for long-term care staff," *Journal of continuing education in nursing*, vol. 33, no. 1, pp. 39–45, 2001.

[46] S. A. Engum, P. Jeffries, and L. Fisher, "Intravenous catheter training system: Computer-based education versus traditional learning methods," *The American Journal of Surgery*, vol. 186, no. 1, pp. 67–74, 2003.

[47] N. D. Cassavaugh and A. F. Kramer, "Transfer of computer-based training to simulated driving in older adults," *Applied Ergonomics*, vol. 40, no. 5, pp. 943–952, 2009.

[48] J. M. Koonce and W. J. Bramble Jr, "Personal computer-based flight training devices," *The international journal of aviation psychology*, vol. 8, no. 3, pp. 277–292, 1998.

[49] S. L. Dawson, S. Cotin, D. Meglan, D. W. Shaffer, and M. A. Ferrell, "Equipment and Technology - Designing a Computer-Based Simulator for Interventional Cardiology Training," *Catheterization and Cardiovascular Interventions*, vol. 51, no. 4, pp. 522–527, 2000.

[50] R. Sims, "Futures for computer-based training: Developing the learner-computer interface," *Australian Journal of Educational Technology*, vol. 4, no. 2, p. 11, 1988.

[51] G. D. Wagner and D. D. Flannery, "A quantitative study of factors affecting learner acceptance of a computer-based training support tool," *Journal of European Industrial Training*, vol. 28, no. 5, pp. 383–399, 2004.

[52] M. Grgurović, C. A. Chapelle, and M. C. Shelley, "A meta-analysis of effectiveness studies on computer technology-supported language learning," *ReCALL*, vol. 25, no. 02, pp. 165–198, 2013.

[53] K. VanLehn, "The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems," *Educational Psychologist*, vol. 46, no. 4, pp. 197–221, 2011.

[54] J. M. Schraagen, S. F. Chipman, and V. L. Shalin, *Cognitive task analysis.* Psychology Press, 2000.

[55] P. N. Blanchard, *Effective Training, Systems, Strategies, and Practices.* Pearson Education India, 4 ed., 1999.

[56] B. Geber, "Does your training make a difference? Prove it," *Training*, vol. 32, no. 3, pp. 27–34, 1995.

[57] J. J. Phillips, *Handbook of training evaluation and measurement methods.* Routledge, 1997.

[58] P. Bramley, *Evaluating training.* Universities Press, 1997.

[59] B. R. Worthen and J. R. Sanders, "Educational evaluation: Alternative approaches and practical guidelines," 1987.

[60] D. Kirkpatrick, "Techniques for evaluating training programs," *Journal of ASTD*, vol. 11, pp. 1–11, 1959.

[61] A. P. Carnevale, L. J. Gainer, and J. Villet, *Training in America: The organization and strategic role of training.* Jossey-Bass, 1990.

[62] J. Holcomb, *Make training worth every penny.* Wharton, 1993.

[63] F. A. McMahon and E. M. Carter, *The great training robbery: A guide to the purchase of quality training.* Falmer Press, 1990.

[64] F. H. E. Rossi, P.H. and S. R. Wright, *Evaluation: A systematic approach.* Sage, 1979.

[65] R. A. Bjork, "Memory and metamemory considerations in the training of human beings," 1994.

[66] L. J. Cronbach and R. E. Snow, *Aptitudes and instructional methods: A handbook for research on interactions.* Ardent Media, 1981.

[67] S. I. Tannenbaum, J. E. Methieu, J. A. Cannon-Bowers, and E. Salas, "Factors that influence training effectiveness: A conceptual model and longitudinal analysis," tech. rep., DTIC Document, 1993.

[68] R. A. Bjork, "Assessing our own competence: Heuristics and illusions," 1999.

[69] R. Christina and R. Bjork, "Optimizing long-term retention and transfer," *In the minds eye: Enhancing human performance*, pp. 23–56, 1991.

[70] L. Jacoby, R. Bjork, and C. Kelley, "Illusions of comprehension, competence, and remembering," *Learning, remembering, believing: Enhancing human performance*, pp. 57–80, 1994.

[71] D. A. Simon and R. A. Bjork, "Metacognition in motor learning.," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 27, no. 4, p. 907, 2001.

[72] R. A. Schmidt and T. Lee, *Motor Control and Learning.* Human Kinetics, 5 ed., 1988.

[73] E. C. Tolman and C. H. Honzik, "Introduction and removal of reward, and maze performance in rats," *University of California Publications in Psychology*, 1930.

[74] J. A. Adams and B. Reynolds, "Effect of shift in distribution of practice conditions following interpolated rest," *Journal of Experimental Psychology*, vol. 47, no. 1, p. 32, 1954.

[75] E. F. Holton, "The flawed four-level evaluation model," *Human resource development quarterly*, vol. 7, no. 1, pp. 5–21, 1996.

[76] E. F. Holton, "Holton's evaluation model: new evidence and construct elaborations," *Advances in Developing Human Resources*, vol. 7, no. 1, pp. 37–54, 2005.

[77] K. Kraiger, *Decision-based evaluation.* Jossey-Bass, 2002.

[78] D. S. Bushnell, "Input, process, output: A model for evaluating training.," *Training and Development Journal*, vol. 44, no. 3, pp. 41–43, 1990.

[79] J. Fitz-Enz, "Yes... you can weigh training's value," *Training*, vol. 31, 1994.

[80] C. P. Van Der Vleuten, "The assessment of professional competence: developments, research and practical implications," *Advances in Health Sciences Education*, vol. 1, no. 1, pp. 41–67, 1996.

[81] R. E. Redding, J. R. Cannon, and T. L. Seamster, "Expertise in air traffic control (ATC): What is it, and how can we train for it?," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 36, pp. 1326–1330, SAGE Publications, 1992.

[82] D. I. Newble and K. Jaeger, "The effect of assessments and examinations on the learning of medical students," *Medical education*, vol. 17, no. 3, pp. 165–171, 1983.

[83] W. J. Popham, "Measurement as an instructional catalyst," *New Directions for Testing and Measurement*, 1983.

[84] N. Frederiksen, "The real test bias: Influences of testing on teaching and learning," *American Psychologist*, vol. 39, no. 3, p. 193, 1984.

[85] N. J. Entwistle, *Styles of learning and teaching: An integrated outline of educational psychology for students, teachers and lecturers.* Routledge, 2013.

[86] P. L. Stillman and D. B. Swanson, "Ensuring the clinical competence of medical school graduates through standardized patients," *Archives of Internal Medicine*, vol. 147, no. 6, p. 1049, 1987.

[87] G. Gibbs, "Improving the quality of student learning through course design," *Learning to effect*, pp. 149–165, 1992.

[88] B. A. Spellman and R. A. Bjork, "When predictions create reality: Judgments of learning may alter what they are intended to assess," *Psychological Science*, vol. 3, no. 5, pp. 315–316, 1992.

[89] W. J. Popham, K. L. Cruse, S. C. Rankin, P. D. Sandifer, and P. L. Williams, "Measurement-Driven Instruction: It's On the Road," *Phi Delta Kappan*, vol. 66, no. 9, pp. 628–34, 1985.

[90] C. H. McGuire and D. Babbott, "Simulation technique in the measurement of problem-solving skills," *Journal of Educational Measurement*, vol. 4, no. 1, pp. 1–10, 1967.

[91] K. Hodgkin and J. D. Knox, *Problem-centred Learning: The Modified Essay Question in Medical Education: a Handbook for Students, Teachers, and Trainers.* Churchill Livingstone, 1975.

[92] G. I. Feletti, N. A. Saunders, and A. J. Smith, "Comprehensive assessment of final-year medical student performance based on undergraduate programme objectives," *The Lancet*, vol. 322, no. 8340, pp. 34–37, 1983.

[93] R. K. Reznick and K. Rajaratanam, "Performance-based assessment," *Teaching and Learning in Medical and Surgical Education: Lessons Learned from the 21st Century. Mahwah, NJ: Lawrence Erlbaum*, pp. 237–43, 2000.

[94] E. R. Petrusa, "Clinical performance assessments," in *International handbook of research in medical education*, pp. 673–709, Springer, 2002.

[95] B. E. Clauser and L. W. Schuwirth, "The use of computers in assessment," in *International handbook of research in medical education*, pp. 757–792, Springer, 2002.

[96] D. Newble, "Techniques for measuring clinical competence: objective structured clinical examinations," *Medical education*, vol. 38, no. 2, pp. 199–203, 2004.

[97] A. Rothman and R. Cohen, "Understanding the objective structured clinical examination (OSCE): Issues and options," *Annals of the Royal College of Physicians and Surgeons of Canada*, vol. 28, pp. 283–287, 1995.

[98] M. J. Govaerts, C. P. Van der Vleuten, L. W. Schuwirth, and A. M. Muijtjens, "Broadening perspectives on clinical performance assessment: Rethinking the nature of in-training assessment," *Advances in Health Sciences Education*, vol. 12, no. 2, pp. 239–260, 2007.

[99] S. G. Pickering, "Against multiple choice questions," *Medical Teacher*, vol. 1, no. 2, pp. 84–86, 1979.

[100] D. Newble, A. Baxter, and R. Elmslie, "A comparison of multiple-choice tests and free-response tests in examinations of clinical competence," *Medical Education*, vol. 13, no. 4, pp. 263–268, 1979.

[101] C. McGuire, "Perspectives in assessment," *Academic Medicine*, vol. 68, no. 2, pp. S3–8, 1993.

[102] L. Schuwirth and C. Van der Vleuten, "The use of clinical simulations in assessment," *Medical Education*, vol. 37, no. s1, pp. 65–71, 2003.

[103] A. S. Elstein, L. S. Shulman, and S. A. Sprafka, *Medical problem solving: an analysis of clinical reasoning*, vol. 2. Harvard University Press Cambridge, MA, 1978.

[104] T. Bligh, "Written simulation scoring: A comparison of nine systems," Proceedings of the American Educational Research Association Annual Meeting, 1980.

[105] D. B. Swanson, J. J. Norcini, and L. J. Grosso, "Assessment of clinical competence: Written and computer-based simulations," *Assessment and Evaluation in Higher Education*, vol. 12, no. 3, pp. 220–246, 1987.

[106] G. Bordage and G. Page, "An alternative approach to PMPs: The "key features" concept," *Further developments in assessing clinical competence*, pp. 59–75, 1987.

[107] E. d. Graaff, G. Post, and M. Drop, "Validation of a new measure of clinical problem-solving," *Medical Education*, vol. 21, no. 3, pp. 213–218, 1987.

[108] D. A. Sloan, M. B. Donnelly, R. W. Schwartz, and W. E. Strodel, "The objective structured clinical examination. the new gold standard for evaluating postgraduate clinical performance," *Annals of Surgery*, vol. 222, no. 6, p. 735, 1995.

[109] V. D. Vleuten, A. Scherpbier, D. Dolmans, L. Schuwirth, G. Verwijnen, and H. Wolfhagen, "Clerkship assessment assessed," *Medical Teacher*, vol. 22, no. 6, pp. 592–600, 2000.

[110] J. Littlefield, D. DaRosa, K. Anderson, R. Bell, G. Nicholas, and P. Wolfson, "Assessing performance in clerkships," *Acad Med*, vol. 66, pp. 516–8, 1991.

[111] G. L. Noel, J. E. Herbers, M. P. Caplow, G. S. Cooper, L. N. Pangaro, and J. Harvey, "How well do internal medicine faculty members evaluate the clinical skills of residents?," *Annals of Internal Medicine*, vol. 117, no. 9, pp. 757–765, 1992.

[112] J. D. Gray, "Global rating scales in residency education," *Academic Medicine*, vol. 71, no. 1, pp. S55–63, 1996.

[113] C. van Barneveld, "The dependability of medical students performance ratings as documented on in-training evaluations," *Academic Medicine*, vol. 80, no. 3, pp. 309–312, 2005.

[114] Federal Aviation Administration, "Advisory Circular 120-51E: Crew Resource Management Training," tech. rep., 2004.

[115] Federal Aviation Administration, "Advisory Circular 120-35C: Line Operational Simulations: Line Oriented Flight Training, Special Purpose Operational Training, Line Operational Evaluation," tech. rep., 2004.

[116] Federal Aviation Administration, "Advisory Circular 120-94: Aircraft Electrical Wiring Interconnection Systems Training Program," tech. rep., 1997.

[117] Federal Aviation Administration, "Advisory Circular 120-72: Maintenance Resource Management Training," tech. rep., 2000.

[118] A. T. Lee and S. R. Bussolari, "Flight simulator platform motion and air transport pilot training," *Aviation, Space, and Environmental Medicine*, 1989.

[119] S. Nählinder, *Flight simulator training: Assessing the potential.* PhD thesis, Department of Management and Engineering, Linköpings University, 2009.

[120] JAA Administrative and Guidance Material, "Section five Part two: Procedures," tech. rep., Joint Aviation Authorities, 2007.

[121] E. Salas, K. A. Wilson, C. S. Burke, and D. C. Wightman, "Does crew resource management training work? an update, an extension, and some critical needs," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 48, no. 2, pp. 392–412, 2006.

[122] R. Flin and L. Martin, "Behavioral markers for crew resource management: A review of current practice," *The International Journal of Aviation Psychology*, vol. 11, no. 1, pp. 95–118, 2001.

[123] R. W. Schvaneveldt, F. T. Durso, T. E. Goldsmith, T. J. Breen, N. M. Cooke, R. G. Tucker, and J. C. De Maio, "Measuring the structure of expertise," *International Journal of Man-Machine Studies*, vol. 23, no. 6, pp. 699–728, 1985.

[124] J. A. Swets and R. A. Bjork, "Enhancing human performance: An evaluation of new age techniques considered by the us army," *Psychological Science*, vol. 1, no. 2, pp. 85–96, 1990.

[125] J. Fowlkes, D. J. Dwyer, R. L. Oser, and E. Salas, "Event-based approach to training (ebat)," *The International Journal of Aviation Psychology*, vol. 8, no. 3, pp. 209–221, 1998.

[126] J. Foulds and E. Frank, "A review of multi-instance learning assumptions," *The Knowledge Engineering Review*, vol. 25, no. 01, pp. 1–25, 2010.

[127] T. M. Mitchell, *Machine learning*, vol. 45. Burr Ridge, IL: McGraw Hill, 1997.

[128] F. Bação, V. Lobo, and M. Painho, "Self-organizing maps as substitutes for k-means clustering," in *Computational Science–ICCS 2005*, pp. 476–483, Springer, 2005.

[129] A. M. Martínez and A. C. Kak, "Pca versus lda," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228–233, 2001.

[130] S. Menard, *Applied logistic regression analysis*, vol. 106. Sage, 2002.

[131] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[132] C. Romero and S. Ventura, "Educational data mining: A survey from 1995 to 2005," *Expert Systems with Applications*, vol. 33, no. 1, pp. 135–146, 2007.

[133] R. S. Baker and K. Yacef, "The state of educational data mining in 2009: A review and future visions," *JEDM-Journal of Educational Data Mining*, vol. 1, no. 1, pp. 3–17, 2009.

[134] C. Romero and S. Ventura, "Educational data mining: a review of the state of the art," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 40, no. 6, pp. 601–618, 2010.

[135] F. Castro, A. Vellido, À. Nebot, and F. Mugica, "Applying data mining techniques to e-learning problems," in *Evolution of teaching and learning paradigms in intelligent environment*, pp. 183–221, Springer, 2007.

[136] J. Mostow and J. Beck, "Some useful tactics to modify, map and mine data from intelligent tutors," *Natural Language Engineering*, vol. 12, no. 2, pp. 195–208, 2006.

[137] C. M. Bingham and R. Crampton, "A review of prevocational medical trainee assessment in New South Wales," *Medical Journal of Australia*, vol. 195, no. 7, pp. 410–412, 2011.

[138] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani, *The elements of statistical learning*, vol. 2. Springer, 2009.

[139] B. Efron, "The bootstrap and modern statistics," *Journal of the American Statistical Association*, vol. 95, no. 452, pp. 1293–1296, 2000.

[140] D. Cervone, A. D'Amour, L. Bornn, and K. Goldsberry, "POINTWISE: Predicting Points and Valuing Decisions in Real Time with NBA Optical Tracking Data," in *8th Annual MIT Sloan Sports Analytics Conference*, 2014.

[141] Federal Aviation Administration, "AFS-210: Line Operational Simulations: Line-Oriented Flight Training, Special Purpose Operational Training, Line Operational Training," tech. rep., US Department of Transportation, 1990.

[142] Civil Aviation Authority, "CAP 720: Flight Crew Training: Cockpit Resource Management (CRM) and Line-Oriented Flight Training (LOFT)," tech. rep., UK Civil Aviation Authority, 2002.

[143] K. Stephens-Martinez, M. A. Hearst, and A. Fox, "Monitoring MOOCs: Which information sources do instructors value?," in *Proceedings of the first ACM Conference on Learning at Scale*, pp. 79–88, ACM, 2014.

[144] A. K. Jain and R. C. Dubes, *Algorithms for clustering data.* Prentice-Hall, Inc., 1988.

[145] L. B. Resnick and D. P. Resnick, "Assessing the thinking curriculum: New tools for educational reform," in *Changing Assessments*, pp. 37–75, Springer, 1992.

[146] J. M. O'Hara, J. C. Higgins, W. F. Stibler, and J. Kramer, "Computer-based procedure systems: technical basis and human factors review guidance," tech. rep., DTIC Document, 2000.

[147] D. Pelleg and A. W. Moore, "X-means: Extending k-means with efficient estimation of the number of clusters," in *Proceedings of theInternational Conference on Machine Learning*, pp. 727–734, 2000.

[148] T. H. Jeff, *Introduction to Neural Networks with Java.* Heaton Research, Inc, 2005.

[149] J. P. Resop, *A comparison of artificial neural networks and statistical regression with biological resources applications.* PhD thesis, University of Maryland, 2006.

[150] C. X. Feng and X. F. Wang, "Surface roughness predictive modeling: neural networks versus regression," *IIE Transactions*, vol. 35, no. 1, pp. 11–27, 2003.

[151] X. W. Chen and J. C. Jeong, "Minimum reference set based feature selection for small sample classifications," in *Proceedings of the 24th international conference on Machine learning*, pp. 153–160, ACM, 2007.

[152] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[153] G. E. Batista and M. C. Monard, "An analysis of four missing data treatment methods for supervised learning," *Applied Artificial Intelligence*, vol. 17, no. 5-6, pp. 519–533, 2003.

[154] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-supervised learning*, vol. 2. MIT press Cambridge, 2006.

[155] R. L. Helmreich, "Culture and error in space: Implications from analog environments," *Aviation Space and Environmental Medicine*, vol. 71, no. 9; PART 2, pp. A133–A139, 2000.

[156] D. A. Norman, "Categorization of action slips," *Psychological Review*, vol. 88, no. 1, p. 1, 1981.

[157] J. Reason, *Human error*. Cambridge University Press, 1990.

[158] E. M. Altmann and J. G. Trafton, "Memory for goals: An activation-based model," *Cognitive Science*, vol. 26, no. 1, pp. 39–83, 2002.

[159] J. G. Trafton, E. M. Altmann, and R. M. Ratwani, "A memory for goals model of sequence errors," *Cognitive Systems Research*, vol. 12, no. 2, pp. 134–143, 2011.

[160] A. Degani and E. L. Wiener, "Procedures in complex systems: The airline cockpit," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 27, no. 3, pp. 302–312, 1997.

[161] D. English and R. J. Branaghan, "An empirically derived taxonomy of pilot violation behavior," *Safety Science*, vol. 50, no. 2, pp. 199–209, 2012.

[162] S. J. Landry and J. A. Jacko, "Improving pilot procedure following using displays of procedure context," *International Journal of Applied Aviation Studies*, vol. 6, no. 1, p. 47, 2006.

[163] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," in *Doklady Physics* (V. E. Fortov, ed.), vol. 10, p. 707, 1966.

[164] D. García-García, E. Parrado Hernández, and F. Díaz-de María, "A new distance measure for model-based sequence clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 7, 2009.

[165] U. Manber and G. Myers, "Suffix arrays: a new method for on-line string searches," *Siam Journal on Computing*, vol. 22, no. 5, pp. 935–948, 1993.

[166] A. Schaafstal and J. Schraagen, "Training of troubleshooting: A structured, task analytical approach," *Cognitive Task Analysis*, pp. 57–70, 2000.

[167] D. Sankoff, G. Leduc, N. Antoine, B. Paquin, B. F. Lang, and R. Cedergren, "Gene order comparisons for phylogenetic inference: Evolution of the mitochondrial genome," vol. 89, pp. 6575–6579, National Academy of Sciences, 1992.

[168] F. J. Damerau, "A technique for computer detection and correction of spelling errors," *Communications of the Association for Computing Machinery*, vol. 7, no. 3, pp. 171–176, 1964.

[169] K. Malde, E. Coward, and I. Jonassen, "Fast sequence clustering using a suffix array algorithm," *Bioinformatics*, vol. 19, no. 10, pp. 1221–1226, 2003.

[170] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," vol. 77, pp. 257–286, Institute of Electrical and Electronics Engineers, 1989.

[171] M. Stengel, "Introduction to graphical models, hidden markov models and bayesian networks," *Department of Information and Computer Sciences Toyohashi University of Technology Toyohashi*, pp. 441–8580, 2003.

[172] A. Panuccio, M. Bicego, and V. Murino, "A hidden markov model-based approach to sequential data clustering," in *Structural, Syntactic, and Statistical Pattern Recognition*, pp. 734–743, Springer, 2002.

[173] R. E. Kass and L. Wasserman, "A reference bayesian test for nested hypotheses and its relationship to the schwarz criterion," *Journal of the American Statistical Association*, vol. 90, no. 431, pp. 928–934, 1995.