9-1-2002

Automatic Target Recognition Classification System Evaluation Methodology

Christopher Brian Bassham

AUTOMATIC TARGET RECOGNITION CLASSIFICATION

SYSTEM EVALUATION METHODOLOGY

DISSERTATION

C. Brian Bassham, Captain, USAF

AFIT/DS/ENS/02-03

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

Report Documentation Page

Report Date	**Report Type**	**Dates Covered (from... to)**
12 Sep 02	Final	Mar 99 - Aug 02

Title and Subtitle Automatic Target Recognition Classification System Evaluation Methodology	**Contract Number**
	Grant Number
	Program Element Number
Author(s) Capt Christopher B. Bassham, USAF	**Project Number**
	Task Number
	Work Unit Number
Performing Organization Name(s) and Address(es) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 P Street, Bldg 640 WPAFB OH 45433-7765	**Performing Organization Report Number** AFIT/DS/ENS/02-03
Sponsoring/Monitoring Agency Name(s) and Address(es) AFRL/Sensor Directorate ATTN: James D. Leonard 2241 Avionics C! Bldg 620 RM N3-X19 WPAFB OH 45433 ACC/DRSA (Combat ID) ATTN: Maj Stewart De Vilbiss 204 Dodd Blvd. Langley AFB, VA 23665	**Sponsor/Monitor’s Acronym(s)**
	Sponsor/Monitor’s Report Number(s)

Distribution/Availability Statement
Approved for public release, distribution unlimited

Supplementary Notes
The original document contains color images.

Abstract

This dissertation research makes contributions towards the evaluation of developing Automatic Target Recognition (ATR) technologies through the application of decision analysis (DA) techniques. ATR technology development decisions should rely not only on the measures of performance (MOPs) associated with a given ATR classification system (CS), but also on the expected measures of effectiveness (MOEs). The purpose of this research is to improve the decision-makers in the ATR Technology development. A decision analysis framework that allows decision-makers in the ATR community to synthesize the performanece measures, costs, and characteristics of each ATR system with the preferences and values of both the evaluators and the warfighters is developed. The inclusion of the warfighter’s perspective is important in that it has been proven that basing ATR CS comparisons solely upon performance characteristics does not ensure superior operational effectiveness. The methodology also captures the relationship between MOPs and MOEs via a combat model. An example scenario demonstrates how ATR CSs may be compared. Sensitivity analysis is performed to demonstrate the robustness of the MOP to value score and MOP to MOE translations. A multinomial section procedure is introduced to account for the random nature of the MOP estimates.

Subject Terms automatic target recognition, decision analysis, performance comparison, multinomial selection procedure, measures of performance, measures of effectiveness, Decision Analysis, comparison, value, utility, ROC curve, evaluation	
Report Classification unclassified	**Classification of this page** unclassified
Classification of Abstract unclassified	**Limitation of Abstract** UU
Number of Pages 227	

The views expressed in this dissertation are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/DS/ENS/02-03

AUTOMATIC TARGET RECOGNITION CLASSIFICATION SYSTEM EVALUATION METHODOLOGY

DISSERTATION

Presented to the Faculty

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

C. Brian Bassham, BS, MS

Captain, USAF

September 2002

AFIT/DS/ENS/02-03

AUTOMATIC TARGET RECOGNITION CLASSIFICATION

SYSTEM EVALUATION METHODOLOGY

C. Brian Bassham, BS, MS
Captain, USAF

Approved:

Date:

__/s/________________________________ ____________
Kenneth W. Bauer, Jr. (Chairman)

__/s/________________________________ ____________
William P. Baker (Dean's Representative)

__/s/________________________________ ____________
John O. Miller (Member)

__/s/________________________________ ____________
Mark E. Oxley (Member)

Accepted:

___/s/______________________________ ____________
Robert A. Calico, Jr. Date
Dean, Graduate School of Engineering and Management

Acknowledgements

I would like to thank several people for their direction and support. First of all, I would like to thank the members of my committee. I especially thank my research advisor, Dr. Ken Bauer, who always kept a big picture of where I was going and served as a huge encouragement. I appreciate the help of Lt Col J.O. Miller and Dr. Mark Oxley. All three made invaluable contributions towards my research.

In regards to my sponsors, I would like to thank Mr. Chuck Sadowski for his enthusiastic contributions, quick responses, and interest. He was truly a gift from heaven during the whole process! I thank Dr. Tim Ross for getting this research going and to Mr. Jim Leonard for continuing AFRL's support.

I cannot thank my wife and two boys enough for their patience and love. This research was extremely difficult because my two boys are such inviting and fun distractions! Most of all, I wish to thank God who let me stay at AFIT and pursue this degree. There are events in my life where I am undeniably certain that God was at work…this opportunity is one of them.

Brian Bassham

Table of Contents

Page

Page

Page

Page

List of Figures

Figure Page

Figure Page

Figure Page

Figure Page

List of Tables

Table Page

Table Page

AFIT/DS/ENS/02M-03

Abstract

This dissertation research makes contributions towards the evaluation of developing Automatic Target Recognition (ATR) technologies through the application of decision analysis (DA) techniques. ATR technology development decisions should rely not only on the measures of performance (MOPs) associated with a given ATR classification system (CS), but also on the expected measures of effectiveness (MOEs) of the potential end product.

The purpose of this research is to improve the decision-making process for ATR technology development. The basis of the research is a decision analysis framework that allows decision-makers in the ATR community to synthesize the performance measures, costs, and characteristics of each ATR system with the preferences and values of both the ATR evaluators and the warfighters. The inclusion of the warfighter's perspective in the decision-making process is important in that it has been proven that basing ATR CS comparisons solely upon performance characteristics does not necessarily ensure superior operational effectiveness. The process for constructing an evaluator and warfighter DA framework is described. The methodology also provides a means for capturing the relationship between MOPs and MOEs via a combat model. An example scenario demonstrates how ATR CSs may be compared. Sensitivity analysis is performed to demonstrate the robustness of the MOP to value score and MOP to MOE translations. A multinomial selection procedure is introduced to account for the random nature of the MOP estimates. Finally, research contributions and future directions are highlighted.

AUTOMATIC TARGET RECOGNITION CLASSIFICATION SYSTEM EVALUATION METHODOLOGY

I. Introduction

1.1 General Discussion

This dissertation research makes contributions towards the evaluation of automatic target recognition (ATR) classification systems (CSs). Though ATR technology has been under development for quite some time, ATR CS evaluation theory is in its infancy. No generally accepted methodology exists for evaluating multiple ATR CSs for the sake of comparison. One aspect of the difficulty lies in the magnitude of the associated set of performance measures, costs, and system characteristics for each ATR CS. This set is often overwhelming and generally leads to a non-dominating solution within ATR CS comparisons. Thus, it is necessary to fuse the subjective preferences of the various decision-makers with the objective realizations of the given performance measures, costs, and system characteristics when making decisions which affect the lifecycle of ATR technology development.

1.2 Motivation

1.2.1 ATR Technology Evaluation Interest.

Improving ATR technology evaluation is of interest to the Sensors Directorate of the Air Force Research Laboratory (AFRL/SN) for its application to unsolved problems associated within the Deputy Under Secretary of Defense (DUSD) Benchmarking (DUSD-BM) program. The decision situation involves the evaluation of several ATR CSs, each having multiple performance measures with which to compare, throughout the life cycle of the DUSD-BM program. Decisions made throughout the program depend on the ability to evaluate a single CS or to compare the performance of several CSs in a fair and useful manner.

This research is also of interest to the Requirements Directorate of Air Combat Command (HQ ACC/DRSA), which is determining the role and applications of Combat Identification (CID) systems. Though not focused solely on ATR technologies, evaluators developing the Targets Under Trees (TUT) program are interested in ascertaining the preferences of the warfighter, which is an important part of this research. Finding the relationship between ATR measures of performance (MOPs) and operational measures of effectiveness (MOEs) is of great importance to each party.

1.2.2 Current ATR System Development Methodology.

Automatic Target Recognition is a processing problem where a classification system, typically in the form of a pattern recognition and classification algorithm, 'examines' an image to detect and classify objects of interest. Various aerial platforms employ various sensory devices, such as synthetic aperture radar (SAR), forward-looking infrared, millimeter wave, or laser radar systems, to collect images for defined military purposes (56). Whether the mission is reconnaissance or targeting, the capacity and capability of these platforms to produce data is overwhelming. Thus, a need for automatically exploiting the abundance of images grows. Currently, human analysis far exceeds the capabilities of automated ATR systems. It is highly desirable to improve automated ATR capabilities, which would increase analytic capacity in military intelligence systems as well as permit ATR systems to be employed on unmanned platforms. Automatic Target Recognition is widely acknowledged as a critical military capability (20).

The objective of an ATR CS is to use the measurable aspects, or features, of a target object located in an image to automatically detect and correctly classify a specific target type, or class, in a real world environment amid various other objects. An ATR CS analyzes the image to identify particular regions of interest (ROIs) and to then classify whether the ROI is a target or not. Typically, the targets are difficult to separate both from normal environmental objects and objects with target-like signatures found within the image.

ATR CSs generally fall into three classes: statistical pattern recognition, neural networks, or model-based recognition. All ATR CSs have a number of quantifiable evaluation measures, such as ATR CS performance, robustness, estimate accuracy, employment doctrine, and cost. In general, these measures are not assessed in total, but specific measures are selected when considering decisions for a specific program. Of noteworthy importance is the method of dealing with the information concerning the class of targets to be classified. A typical DUSD-BM ATR CS differs from previous ATR approaches in that instead of relying on a vast data library of stored target images, the CS is a model-based approach that uses computer-generated templates for classifying a particular image (7). This method requires a smaller data library of stored target images (7).

Within the ATR research community, a *target* is an object in an image that is to be found, or *discriminated*, by an ATR CS, but not necessarily destroyed (55). Thus, friendly objects, such as the American M1 Abrams main battle tank (MBT), may be labeled as a target in ATR scenarios. *Clutter* refers to objects detected within areas of unknown or untracked objects, such as fields or forests, which presumably contain no targets (2). *Confusers*, on the other hand, are objects similar in size and appearance to the targets that are tracked during ATR testing, but are *not* to be detected by the ATR CS (2). These *non-targets* are used to confuse the CS with a target-like signal. One important concept is that when clutter objects are made known, or *truthed*, during the testing phase, they are considered confuser items and are included in the non-target calculations (55). The performance of an ATR CS against non-target objects provides insight into the algorithm's sensitivity of detection.

The purpose of the DUSD-BM program is to evaluate a wide variety of ATR programs to produce an end product that maximizes operational effectiveness. Several different ATR technologies have been developed through several research programs, including the Moving and Stationary Target Acquisition and Recognition (MSTAR), the Non-Cooperative Target Identification (NCTI) and the Air-to-Ground Imaging (AGRI) programs (55). Since these programs approach similar problems in different ways or approach different problems in the same way, the challenge of the DUSD-BM program is to identify the pertinent MOPs that appeal to the researchers in the field of ATR for their utility and robustness while striving to optimize the associated MOEs of each ATR technology.

1.3 Problem Statement

Concerning the theories behind the objective evaluation of ATR CSs, there are several limitations. The first problem stems from the limitations found when testing ATR systems. Assessment of operational performance is difficult due to the small sample size of enemy systems and the regulations governing testing site operations. Both hinder accurate recreation of operational conditions during ATR testing. Next, the number of performance measures is often difficult to manage. Thus, a subset of MOPs is typically used to evaluate ATR CSs. Also, the objective evaluation of ATR systems via MOPs generally leads to a non-dominating solution. Most importantly, it can be shown that optimizing a set of ATR performance characteristics does not necessarily lead to an optimized solution in terms of operational effectiveness (60).

Consider two ATR systems being compared using the performance data listed in Table 1.1. The two systems (A and B) are based upon a declaration and reporting fusion concept between multiple sensors. System A declares a system when both sensors agree at a target's type level, i.e. both sensors claim that a target is a T-72 main battle tank (MBT). System B uses a fusion concept at the target's Friend/Enemy/Neutral (FEN) level. For System B to declare a target as Friendly or Neutral, it requires both sensors to agree. However, for a target to be declared as Enemy or Ambiguous, at least one sensor needs to declare the target as such. Viewing Table 1.1, it is obvious that one cannot use the MOPs to declare a dominating system based upon performance. While System A has a superior FEN identification confidence rate to System B, it has an inferior FEN declaration rate. In other words, System A has high confidence in its declarations, but it does not make as many declarations as System B. Arguably, System B is the better overall system due to its high declaration percentage for all targets.

Table 1.1 Measures of Performance for Hypothetical ATR Systems (60).

MOPs	System A	System B
FEN Identification Declaration Rate	< 40%	> 60%
FEN Identification Confidence Rate	> 99%	> 90%
Class/Type Identification Declaration Rate	< 40%	0%
Class/Type Identification Confidence Rate	> 99%	N/A
Critical Error Rate	0.1% to 1%	0.6% to 2%
Declaration Percentage for All Targets	39%	63%

However, when the performance characteristics of these systems are introduced to a combat model, the MOEs, listed in Table 1.2, indicate a different conclusion. Operationally, use of System A results in a shorter conflict with fewer allied fatalities, more enemies killed per day, and a lower incidence and rate of fratricide. Thus, simple comparisons based solely upon MOPs are insufficient for measuring ATR operational effectiveness.

Table 1.2 Measures of Effectiveness for Hypothetical ATR Systems (60).

MOEs	System A	System B
Length of Conflict (days)	80	100
Friends Killed (day/conflict)	3 / 240	5 / 500
Neutrals Killed (day/conflict)	0 / 0	0 / 0
Enemy Killed (day/conflict)	25 / 2000	20 / 2000
Fratricide (day/conflict)	0.35 / 28	0.4 / 40
Fratricide Error Rate	1.5%	2.0%

Therefore, the problem with evaluating ATR systems, especially for the sake of comparison, is that MOPs do not directly translate into MOEs and that subjective preferences must be incorporated at some point within the decision-making process. The major thrust of this research is to create a practical evaluation methodology within the ATR technology research and development system that incorporates the subjective preference structure of each decision-maker, includes the objectivity of each pertinent MOP, and exploits the relationship between ATR performance characteristics and operational effectiveness.

1.4 Organization of Dissertation

The remainder of the dissertation is organized as follows. Chapter II provides a literature review of current ATR evaluation theory and techniques. Chapter III introduces the implementation of the decision analysis (DA) techniques and assumptions required for the proposed evaluation methodology. Two separate DA frameworks are constructed: one takes the perspective of the ATR evaluator, the other views evaluation from a warfighter perspective. Data from the MSTAR program illustrates how DA techniques may be implemented when comparing several different ATR systems at the ATR evaluator decision-making level. Chapter III also details how MOPs can be translated into MOEs via a combat model. Chapter IV steps through the process of analyzing the outputs of the two decision frameworks via an application using three notional ATR CSs. Chapter V provides an analysis of the sensitivity of the DA frameworks. A linear regression approach, which utilizes a surrogate combat model, creates a differentiable response surface of the value scores from which the partial differentiations of individual MOPs may be calculated as to identify the salient features of the MOP set. Chapter VI introduces a multinomial selection procedure as a means for a decision-maker to compare several different ATR CSs and make defensible selections by accounting for the inherent variation found within MOP sampling and by associating a level of confidence to the value and utility scores generated through both DA frameworks. Research contributions are summarized and future directions are highlighted in Chapter VII.

II. Literature Review

2.1 Overview

This chapter reviews the pertinent literature on the two main topics required to complete this dissertation research—performance measure assessment and performance measure comparison for ATR CSs. Much of the discussion on performance measure assessment focuses on air-to-ground ATR research, particularly performance measure assessment within the MSTAR program. Though the performance measure names and assessment methods may differ between ATR programs, the general assessment and comparison concepts still apply.

2.2. Automatic Target Recognition Performance Measure Assessment.

2.2.1 Background.

To understand the problem at hand, it is imperative to grasp the concepts and implications of the several measures of performance used within the confines of ATR. Model-based ATR CSs are developed on a given set of training conditions and then tested with a set of testing conditions, which are not comprised entirely of the same training conditions. The evaluation of CS performance lends insight into the possible operational environment performance of the CS. Operational conditions (OCs) for DUSD-BM ATR CSs can be viewed as falling into one of three categories: environmental, sensor, and target (4). Environmental conditions, such as revetments and adjacency to other targets, reflect the various backgrounds and obstructions that will make a target difficult to find. Sensor conditions consist of the variation used in the aerial platforms during tests, such as depression angle to target and sensor gain factor. Target OCs include the multitude of variations that a specific target type may assume. These variations to the target SAR image signature may include the different settings for articulated parts (turrets or doors), various external attachments (fuel drums or tools), or the different mission setting of a specific vehicle (rescue, command post, or personnel carrier) (4).

The objective of evaluation is to measure CS performance by constructing and using valid performance measures. With the limiting test scenarios used for training ATR CSs, there is considerable thought on what the collected performance measures actually measure and what inferences may be drawn from them. Performance measure validity is fulfilled through the following evaluation concepts: accuracy, robustness, extensibility, and utility (54,58). Accuracy refers to the absence of a bias or error of a given CS, under the conditions of its training (54,58). Thus, if a CS is not accurate under test conditions, then it is unlikely that the CS will be accurate outside the training conditions (58). A robust system refers to how well a CS performs outside of the training conditions and outside of its modeled conditions (58). Thus, robustness provides information as to how an ATR CS will perform under operational conditions (58). Extensibility is the performance of a CS outside of the training conditions but within the modeled conditions (58). The extensibility of a CS tells a decision maker valuable information about operational performance by giving insight into the payoff of the model-driven components of the CS. Finally, CS utility is simply the performance of the system under operational conditions (58). Though this is the evaluation concept of most interest and relevance, it is generally the most difficult to obtain (54,58). Figure 2.1 provides an abstract, graphical representation of the relationships between testing, training, operational, and modeled conditions. Figure 2.2 does the same for accuracy, robustness, extensibility, and utility.

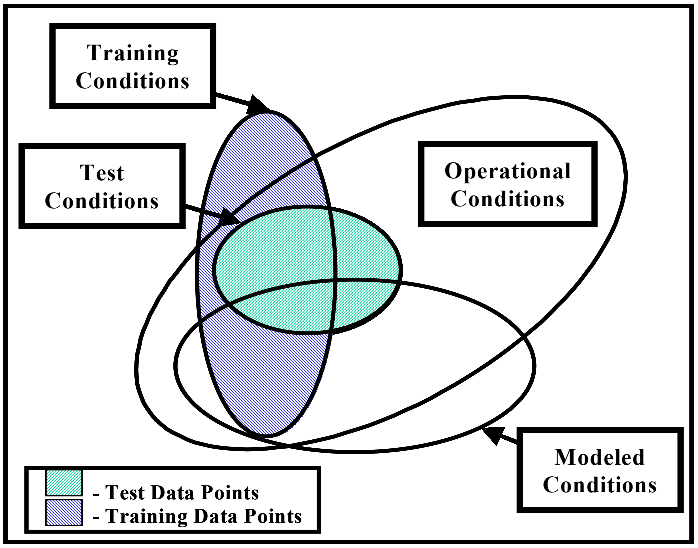


Figure 2.1 Venn Diagram of Modeled, Training, and Testing Conditions (58).

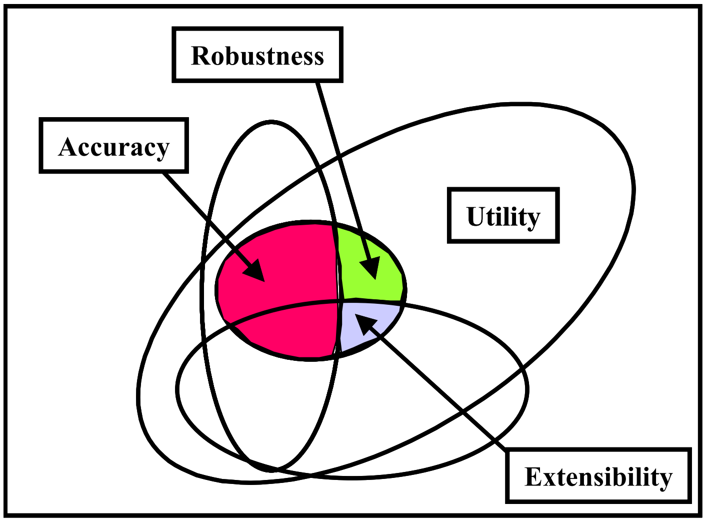


Figure 2.2 Accuracy, Robustness, Utility, and Extensibility (58).

For a generic ATR application, performance is typically assessed using a set of probabilities (3). For the DUSD-BM program, the performance metrics of interest include: the probability of detection (p_D), probability of identification (p_{ID}), probability of correct classification (p_{CC}), and probability of false alarm (p_{FA}). Closely related to p_{FA} is the false alarm rate (*FAR*), which is the ratio of the number of false alarms to the area being examined by the sensor. Another method for assessing CS performance is the use of receiver operating characteristic (ROC) curves. ROC curves are commonly used for summarizing the performance of imperfect diagnostic systems, especially in ATR and biomedical research (7). Finally, new performance measures can be created through the synthesis or manipulation of existing performance measures.

2.2.2 Confusion Matrices.

Possibly the most succinct and popular way of reporting classification results of ATR CSs is the use of confusion matrices, also known as a discrimination event matrix. The matrix is a square grid with a single row and a single column corresponding to each category defined in the data set. The rows refer to the truth membership of each category, while the columns refer to the predicted, or classified, membership. The (*i,j*) cell in the matrix is the number of predicted classifications on category *j* that correspond to the truth source of category *i*. Confusion matrices may also include the relative proportions for each cell by dividing the contents of the cell by the total number of objects that belong to that row, or truth. Another method of presenting a confusion matrix is to combine all targets and non-targets together into a 2x2 composite confusion matrix. Figure 2.3 depicts the form of a standard confusion matrix and identifies several relationships and terms associated with confusion matrices within the fields of pattern recognition, biostatistics, and ATR research. Tables 2.1 and 2.2 provide numerical examples of confusion matrices.

The strengths of confusion matrices are the ability to determine the power of a diagnostic system over the entire data set and to identify where deficiencies are occurring. Confusion matrices, however, do not provide a measure of effectiveness for comparing multiple CSs and only visualize CS performance at a single decision threshold point (7).

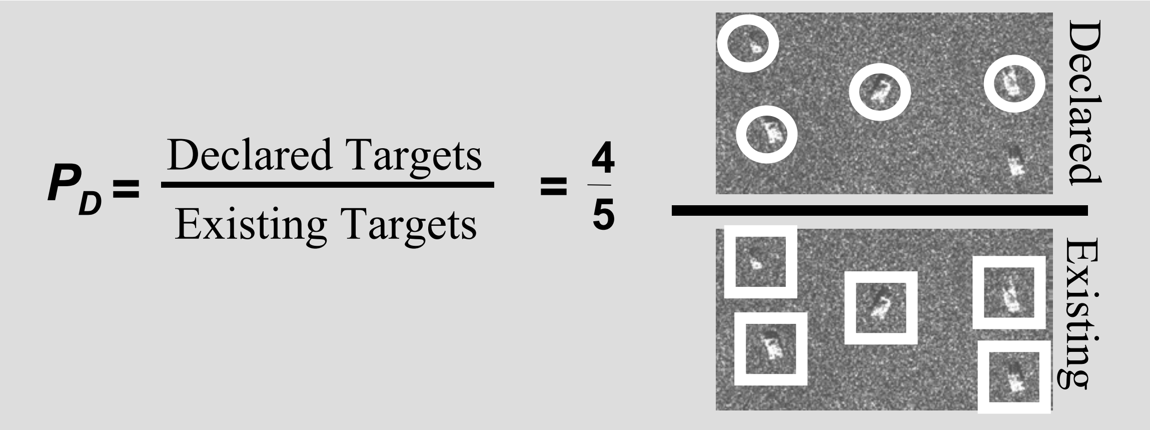
<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="2"><i>Classified As</i>
(Reported As)
(Diagnosed As)</td></tr>
<tr><td>Target
TGT
Abnormal
Diseased
<i>S</i></td><td>Clutter
Non-TGT
Normal
Non-diseased
<i>N</i></td></tr>
<tr><td rowspan="2"><i>Truth</i>
(Known)</td><td><i>Target</i>
TGT
Abnormal
Diseased
<i>sn</i></td><td>True Positives (TP)
Sensitivity
P_{TP}
$P(S|sn)$
p_D
Power
$1-\beta$</td><td>False Negative (FN)
1-Sensitivity
P_{FN}
$P(N|sn)$
$1-p_D$
1-Power
β</td></tr>
<tr><td>Clutter
Non-TGT
Normal
Non-diseased
<i>n</i></td><td>False Positive (FP)
1-Specificity
P_{FP}
$P(S|n)$
p_{FA}
Level of Sig
α</td><td>True Negatives (TN)
Specificity
P_{TN}
$P(N|n)$
$1- p_{FA}$
Confidence
$1-\alpha$</td></tr>
</table>

Figure 2.3 Confusion Matrix with Associated Terms and Relationships.

2.2.3 Probability Performance Measures.

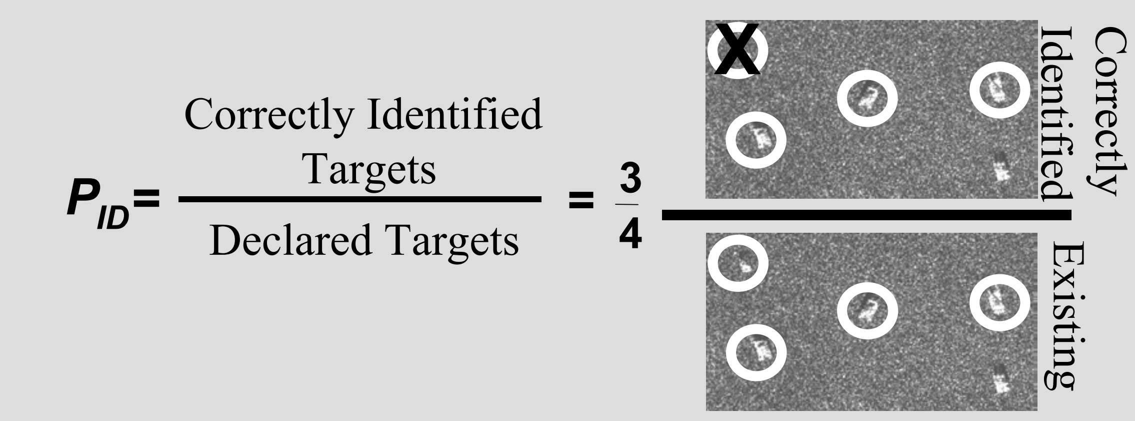
In order to evaluate how an ATR performs in the real world where the target and non-target densities are unknown, use of statistical measures that estimate performance against the known target densities within a testing environment is necessary. Probabilistic performance measures quantify how an ATR CS performs on a given data set. The probabilistic measures detailed in this section are specific to the MSTAR program, but the concepts apply to all DUSD-BM programs.

The utility of a probability of success measure depends on the scenario in which it is used. The most basic of the several probability measures, the probability of detection, designated p_D, is simply the probability that a certain number of the total number of known targets are detected by the ATR in a test scenario (2). AFRL defines *correct detection* as correctly declaring that a target in a region of interest (ROI) is, in fact, a target. Next, the probability of correct classification (p_{CC}) is the probability that an ATR detects a target and associates the target with the appropriate target type. AFRL/SN defines *correct classification* as correctly classifying a detected target as a member of its actual target class regardless of the specific target type. For instance, if the ATR detects what is known to be a T72 MBT and classifies it as a member of the 'tank' class of targets, then the result is a correct classification. Notice that p_{CC} is conditional on whether or not the target is actually detected (2). Thus, the fact that a target is not detected does not factor into the correct classification calculation. The same can be said for the next probability, the probability of *correct identification* (p_{ID}). Correct identification, a subset of correct classification, is when an ATR correctly declares the specific target type of a detected target. Thus, were the ATR to detect the T72 and correctly identify it as a T72 MBT, the result is considered a correct identification. Such events are captured in the p_{ID} metric (2). A final probability measure attempts to capture the number of incorrect decisions, or *false alarms*, made by the ATR. False alarms typically occur when an ATR declares a non-target, or confuser, as a target. The probability of false alarm, p_{FA}, is the ratio of detected non-targets to the total number of known, or *truthed*, non-targets. The following figures provide a visual demonstration of how these probability performance measures are calculated.



$P_D = \frac{\text{Declared Targets}}{\text{Existing Targets}} = \frac{4}{5}$
Declared
Existing

Figure 2.4 Graphical Example of Probability of Detection, p_D (3).



$P_{ID} = \frac{\text{Correctly Identified Targets}}{\text{Declared Targets}} = \frac{3}{4}$
X
Correctly Identified
Existing

Figure 2.5 Graphical Example of Probability of Correct Identification, p_{ID} (3).

$$P_{FA} = \frac{\text{Detected Non-Targets}}{\text{Existing Non-Targets}} = \frac{1}{3}$$

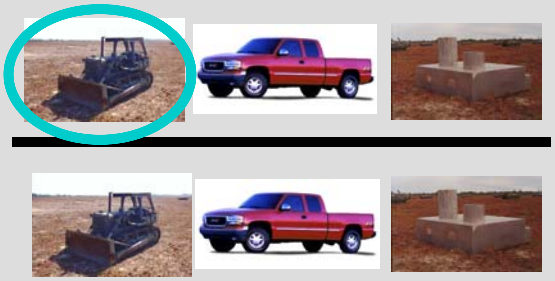


Figure 2.6 Graphical Example of Probability of False Alarm, p_{FA} (3).

A simple, numeric example will provide insight into the calculation of the aforementioned performance measures. In the example, there are 14 known T72 tanks, 14 M1 tanks, 17 Scud missile launchers, and 41 non-targets (confusers). Notice that the T72 and the M1 both belong to the MBT class while the Scud launcher is a member of the Mobile Rocket Launcher System (MRLS) target class. The following confusion matrices provide the fabricated DUSD-BM results.

Table 2.1 Example Confusion Matrix.

Classified As (Reported)

		T72	**M1**	**SCUD**	**Non-Target**
Actual (Truth)	**T72**	12	2	0	0
	M1	2	8	3	1
	SCUD	0	0	7	10
	Non-Target	0	1	5	35

Table 2.2 Example Composite Confusion Matrix.

Classified As (Reported)

		Target	**Non-Target**
Actual (Truth)	**Target**	34 (75.6%)	11 (24.4%)
	Non-Target	6 (14.6%)	35 (85.4%)

Thus, this particular ATR reported that the image contained 14 T72s, 11 M1s, 15 Scud missile launchers, and 46 non-target objects. The following equations provide the calculations of the given performance measures:

$$\hat{p}_D = \frac{number\ of\ ROIs\ declared\ as\ t\arg ets}{number\ of\ known\ \ t\arg et\ objects} = \frac{14+10+10}{14+14+17} = 75.6\% \qquad (2.1)$$

$$\hat{p}_{CC} = \frac{number\ of\ ROIs\ correctly\ classified\ by\ t\arg et\ class}{number\ of\ ROIs\ declared\ as\ t\arg ets} = \frac{14+10+7}{14+10+10} = 91.2\% \quad (2.2)$$

$$\hat{p}_{ID} = \frac{number\ of\ ROIs\ correctly\ identified\ by\ t\arg et\ type}{number\ of\ ROIs\ declared\ as\ t\arg ets} = \frac{12+8+7}{14+10+10} = 79.4\% \quad (2.3)$$

$$\hat{p}_{FA} = \frac{number\ of\ confusers\ declared\ as\ t\arg ets}{number\ of\ known\ confuser\ objects} = \frac{1+5}{41} = 14.6\% \qquad (2.4)$$

While the example illustrates how the performance measures are calculated, there is too little information to gain an overall appreciation of the ATR. For instance, the area covered, terrain type, mission type, or target density is not given. Also, no performance baseline for an acceptable ATR is mentioned. Perhaps in the same setting the average ATR may have a superior performance. Finally, these performance measures are estimates of the true performance measures for the given ATR. The ATR may perform well on such a small data set, but may not over a much larger, more realistic target setting. The practice of placing confidence intervals around these point estimates will be mentioned later in the performance measure evaluation section.

Finally, consideration should be given to the use of probabilistic performance measures between ATR studies. There are several different ways to compute and label these measures since there are several different ATR programs with differing objectives.

For instance, the NCTI ATR program incorporates the use of a probabilistic measure called p_{ND}, since NCTI classifiers also declare a *No Decision* category in addition to the target and non-target declaration decision (3). Also, DUSD-BM performance measures, like p_{CC} and p_{ID}, are conditional on the detection results of the CS, while NCTI metrics are not conditional (3). Therefore, although this large variety of measures can be reduced to a small independent set, caution must be used when comparing probabilistic measures across ATR studies.

2.2.4 Rate Measures.

A common measure used in ATR evaluation is called the false alarm rate (FAR). This rate is merely the number of false alarms in clutter divided by the area of the imagery evaluated by the ATR (2). This measure offers a glimpse into the clutter density of a given area and the propensity of a given ATR CS to detect non-target objects. Figure 2.7 depicts how FAR is typically calculated. The different uses of this metric change the ways in which it is measured. For instance, when measuring the FAR for forward-looking infrared data, the FAR may be computed as false alarms per frame or per second (55). The strength of the FAR metric is that it is transportable to the area to be observed. In other words, if the FAR of a desert-like environment is quantified through testing, then when an ATR CS performs in a previously untested desert location, the FAR should be similar to the estimated FAR (55). In fact, since they rely on environmental and sensor aspects rather than scenario and target assumptions, estimated FARs are likely to be the most reliable and operationally sound metrics that the AFRL COMPASE Center have gathered (55). Since the p_{FA} and FAR performance are related and much of the literature, particularly the extensive medical literature, focuses on p_{FA}, the discussion within this report generally uses the p_{FA} metric for false alarm performance.

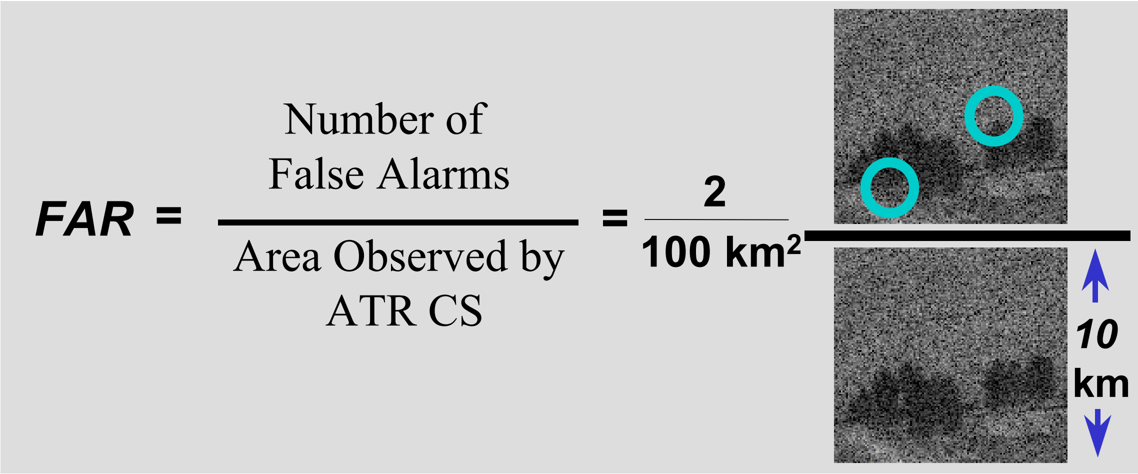


Figure 2.7 Graphical Example of False Alarm Rate, *FAR* (3).

2.2.5 Confidence Intervals.

Metrics such as the probabilistic measures are actually point estimates of a true performance measure of a given CS. The calculation of a single number for a given performance measure is insufficient in that it ignores the amount of data used to produce the quantitative value of the measure. For example, an ATR designer may suggest that a given system may have a p_D of 0.756 (as in Table 2.2). However, since the ATR CS is tested on a finite set of data, the true p_D is probably not 0.756 (7). Instead, the true parameter probably lies within an interval centered about the point estimate. This *confidence interval*, constructed from statistical assumptions and the sample size of the test, allows the ATR designer to make certain inferences about the experimental uncertainty of his CS and its true p_D (7).

The general procedure for constructing a confidence interval is to first postulate an underlying distribution. Distributions that are frequently used in the area of ATR evaluation are the Gaussian, Binomial, and Poisson distributions (7). The following example illustrates how a confidence interval can be obtained about the point estimate of a given performance measure.

Suppose an ATR is designed and tested on an independent sample. For each ROI examined by the ATR CS, there are two separate decision alternatives:

$$\eta = \begin{Bmatrix} 0, & \text{if ROI incorrectly classified} \\ 1, & \text{if ROI correctly classified} \end{Bmatrix} \tag{2.5}$$

with associated probabilities: $P(0) = 1\text{-}p$ and $P(1) = p$. This means that η is a Bernoulli random variable. For a series of independent, identical trials, the Binomial random variable Y is the number of successful classifications in n trials (41). Thus, Y~binomial(n,p) where:

$$p(y) = \binom{n}{y} p^{y} (1-p)^{n-y} \tag{2.6}$$

and the expectation and variance of Y are given by:

$$E(Y) = np \quad \text{and} \quad Var(Y) = np(1-p)\,. \tag{2.7}$$

Using the definition of the probability of detection can make an unbiased estimate for p, the true probability of detection:

$$\hat{p} = \frac{Y}{n}\ . \tag{2.8}$$

Now, $\hat{p}$ is an unbiased estimator for p, so:

$$E(\hat{p}) = E(\frac{Y}{n}) = \frac{1}{n} E(Y) = p\ . \tag{2.9}$$

Using the variance expression for $\hat{p}$ yields:

$$Var(\hat{p}) = Var(\frac{Y}{n}) = \frac{1}{n^2} Var(Y) = \frac{p(1-p)}{n} \tag{2.10}$$

and the usual method of substituting sample values for unknown parameters in the expression for the variance, one can approximate (1-α) confidence intervals for $\hat{p}$ as:

$$\hat{p} \pm Z_{(1-\frac{\alpha}{2})} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \tag{2.11}$$

where the normal approximation is used (assuming large test sample size; $n > 30$) for the binomial. For the example above in Table 2.2 ($p_D = 0.756$ and n = 45), with $\alpha = 0.05$, the following 95% confidence interval is generated for the point estimate:

$$\hat{p} = 0.756 \pm 0.1255 \quad \text{or} \quad 0.6305 \leq \hat{p} \leq 0.8815. \tag{2.12}$$

The strength of confidence interval utilization is that it provides a measure of the accuracy of the evaluation process. Thus, small sample sizes propagate large intervals about a point estimate in which there is little confidence about the location of the true parameter. Large sample sizes tend to narrow the intervals about a point estimate and provide a certain amount of confidence based on the underlying assumptions made about the distribution of the data. Confidence intervals allow the comparison of multiple CSs under the same conditions by quantifying the possible variance in CS performance.

Confidence intervals are limited in that they provide information on how a CS is expected to perform in the future under the same conditions (7). Confidence intervals are not guaranteed to be robust over many different scenarios. Thus, if any of the operational conditions of a given ATR test are changed, the confidence intervals about a performance measure point estimate will not be valid over the new conditions.

2.2.6 Hypothesis Testing.

Hypothesis testing and point estimation form the two branches of classical statistical inference (28). Confidence intervals reveal insight into the strength of a point estimate. Hypothesis testing uses statistical evidence to justify or reject a suspected inference through the strength of a given point estimate. In the typical language of hypothesis testing, the null hypothesis, denoted by H_0, is a statement made about the given point estimate which is being evaluated. For instance, the null hypothesis may state that the p_D of a given ATR CS equals a certain value, e.g. H_0: $p_D = 0.8$. The null hypothesis is usually tested against an alternative hypothesis, H_A., which is typically the opposite of the null hypothesis. Thus, an example of an alternative hypothesis would be H_A: $p_D \neq 0.8$. In order to test a hypothesis, a test of significance approach may be taken. A hypothesis test uses a test statistic based upon a probability distribution. For instance, a test statistic based upon the normal distribution is represented by

$$Z_0 = \frac{\bar{x} - \mu}{\frac{\sigma}{\sqrt{n}}} \approx N(0,1). \tag{2.13}$$

The test statistic, Z_0, may now be compared against the probability distribution of a standardized normal pdf, $N(0,1)$. An acceptance region can be thought of as the portion of the $N(0,1)$ distribution that is suggested by a given acceptance parameter, generally given as an alpha (α) parameter. If the test statistic lies within an acceptance region, the null hypothesis is accepted. Otherwise, the null hypothesis is rejected, and the alternative hypothesis is accepted.

Suppose that a given ATR CS has been tested against 45 known target objects and has correctly classified 34 as targets, resulting in a p_D of 0.756 (using Table 2.2 as data). The goal of hypothesis testing is to determine, given the above information, whether or not the ATR CS meets a designated p_D goal for the system. Further suppose that for an ATR CS to be eligible for evaluation continuance, a p_D of 0.8 must be demonstrated. The p_D value (0.756) of the given CS suggests that this system should be removed from consideration of further study since it falls below 0.8. However, if a hypothesis test is performed, the test statistic Z_0 is formed as:

$$H_0 \; : \; p_D \geq 0.8 \tag{2.14}$$

$$H_A \; : \; p_D < 0.8 \tag{2.15}$$

$$Z_0 = \frac{\hat{p}_D - p_0}{\sqrt{\dfrac{\hat{p}_D(1-\hat{p}_D)}{n}}} \approx N(0,1) \tag{2.16}$$

where the test statistic is based upon the Binomial distribution, and p_0 is the required p_D performance value (0.8). Calculating the test statistic and applying the result to the one-sided hypothesis test (using α=0.1) provides the following:

$$Z_0 = -0.68723 > -Z_A = -1.28. \tag{2.17}$$

Thus, the null hypothesis is accepted (or fails to be rejected), which implies that there is no statistical evidence that the p_D of the ATR CS fails to meet the p_D requirement of 0.8. In fact, a p_D point estimate value as low as 0.71376 would allow the null hypothesis to be accepted. Figure 2.8 depicts the hypothesis rejection region (shaded) and the value of the null hypothesis.

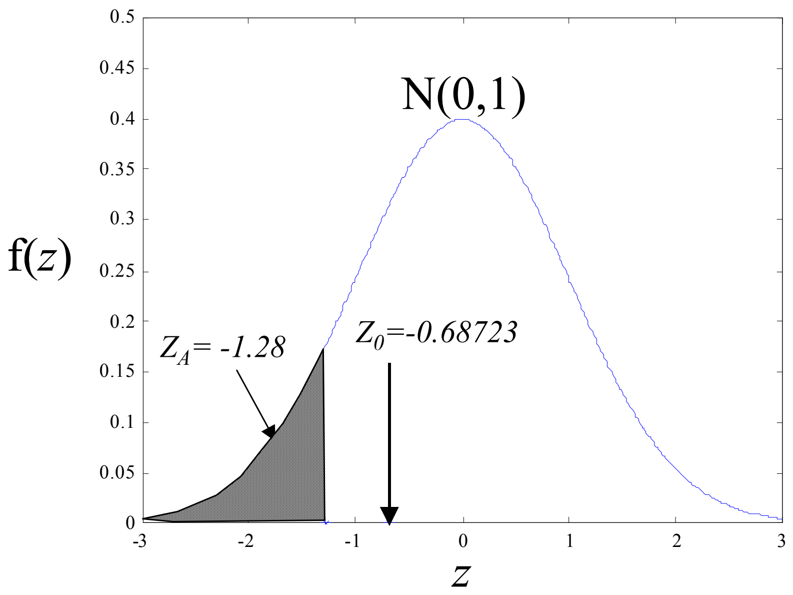


Figure 2.8 Graphical Description of Hypothesis Testing Example (α=0.1).

Hypothesis testing provides a formal approach to the statistical evaluation of CSs. However, its weaknesses are similar to those of confidence intervals. For example, confidence interval and hypothesis testing based on the Binomial distribution assumes a constant probability of success and a constant variance for all observations. However, as an example of when hypothesis testing may not work in an ATR evaluation context, the classification performance of an ATR CS may be different for different targets at different aspect and depression angles. Finally, accepting the null hypothesis does not mean that the hypothesis is correct. The acceptance of a hypothesis simply means that no statistical evidence proves that the hypothesis is false. Accepting a hypothesis is also referred to as *failing to reject* a hypothesis. Therefore, inferences made with hypothesis testing must be used with caution, especially when making inferences about performance measures based on small sample sizes.

2.2.7 Receiver Operating Characteristic (ROC) Curve Performance Measures.

The receiver operating characteristic (ROC) curve is an important technique in summarizing the power of imperfect diagnostic systems that attempt to detect a signal in noise (7,23,27,51,66). An ROC curve describes the relationship between a diagnostic system's sensitivity (probability of selecting true positives) and specificity (selecting true negatives) (66). Though an ROC curve may model the outcomes for multiple alternatives or variable decision rules, only two-alternative, forced-decision (2AFC) ROC curves will be discussed here in the context of ATR CSs (27). Figure 2.9 provides examples of ROC curves for imperfect diagnostic systems. Note that the *D*(*n*,*n*) notation is explained in a subsequent section.

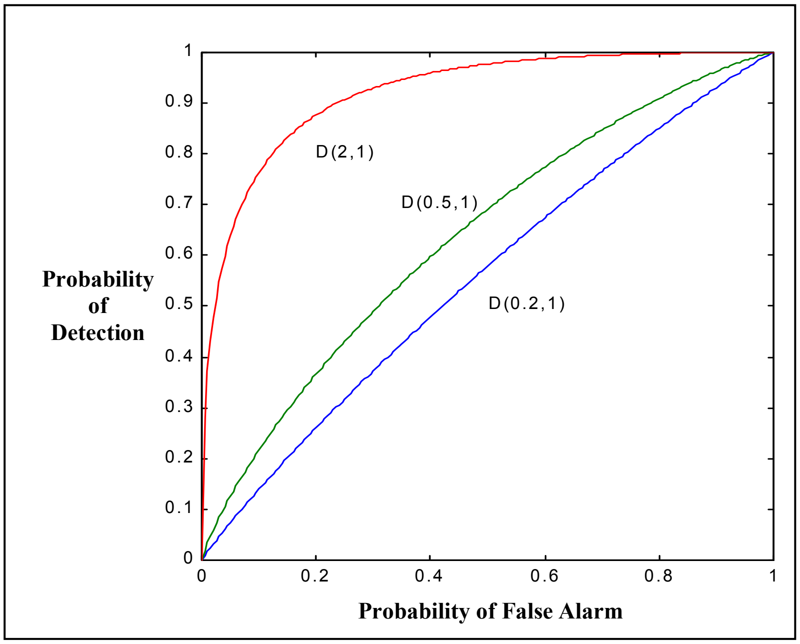


Figure 2.9 Binormal 2AFC ROC Curves.

The two alternatives typically used in an ATR CS decision-making context are the target and non-target classifications. For the following examples, the distribution that represents the target signal will be designated *sn* (signal plus noise) while the non-target signal distribution will be labeled *n* (noise). The term *signal* mentioned here is synonymous with the term *score* in the ATR context. The response of the ATR classifier is restricted to the same alternatives: *S* denotes the event that the classifier reports a target, while *N* denotes the event that the classifier reports a non-target.

Since the objective of the problem at hand is to detect and correctly identify an observed signal (score), most probably degraded by noise, it is natural to investigate the concepts within signal detection theory. Signal detection theory refers to the science behind the "process of detection and recognition of a wanted, or useful, signal that has been degraded by noise" (23). Two separate concepts form the basis of signal detection theory: distribution theory and decision theory. Distribution theory, of which ROC analysis is a subset, refers to the relationships between wanted signal (target) and noise (non-target) distributions. Decision theory refers to the rules used in decision making that hope to satisfy some decision goal. The aspects of decision theory that impact the ATR problem are handled by the ways in which the ATR CS measure and compare individual scores for each ROI.

In terms of distribution theory, the primary challenges to correctly classifying an object as a target or non-target are found in (1) the variability of the target and non-target ROI values (scores) when observed by an ATR sensor and (2) the tendency of the target score distribution to overlap the non-target score distribution, as seen in Figure 2.10. For DUSD-BM ATR CSs, raw data is used to represent the distribution of target scores empirically. However, for ease of explanation, probability distribution functions (pdfs) will be used to represent the score distributions of targets and non-targets. These pdfs are generally assumed to be Gaussian in the medical literature, which is sound theoretically due to the properties of the Central Limit Theorem and sound practically due to the ease of parameter manipulation (27). ROC curves based upon two normal distributions representing the populations of interest are referred to as N-N ROC curves, or binormal ROC curves (23,66). For the examples depicted in Figures 2.10 and 2.11, the non-target pdf is distributed as a $N(0,2)$, and the target pdf is distributed as a $N(4,1.5)$. While the empirical distribution of DUSD-BM ATR target and non-target scores may have a bell-shape similar to Gaussian distributions, the actual scores tend to have more observances farther away from the mean of the distribution, or put simply, the distribution has “fatter” tails. In general, DUSD-BM ATR CSs make no underlying assumption about the distribution of target and non-target score distributions (55).

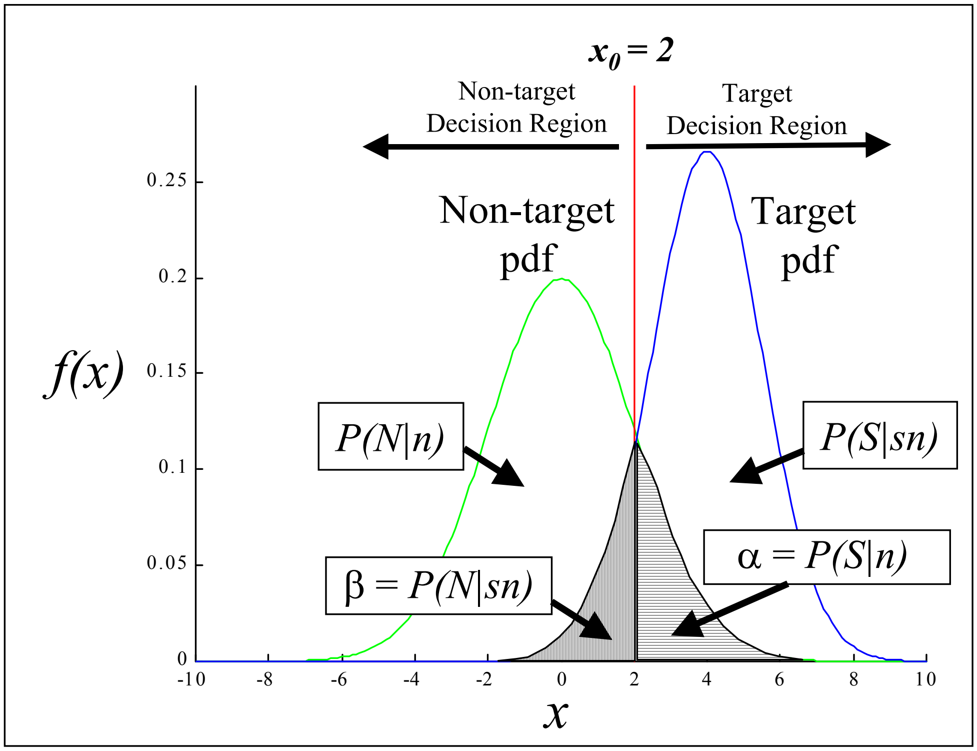


Figure 2.10 Target and Non-target Normal pdfs for a 2AFC Task.

A fundamental concept in decision theory is that of the likelihood ratio function, *L(x)* (23). Where *x* is a specific value of the continuous random variable *X*, the relationship between *L(x)* and *x* summarizes the changing ratio between the corresponding probability densities, *sn* and *n* (23). The likelihood ratio function for continuous target and non-target distributions is:

$$L(x) = \frac{f(x \mid sn)}{f(x \mid n)}, \tag{2.18}$$

and for discrete distributions it is:

$$L(x) = \frac{P(x \mid sn)}{P(x \mid n)}. \tag{2.19}$$

The likelihood ratio is important in that the observer can control the value of the *L(x)* required for a desired probability of target detection (23). An observer could raise/lower the value of the threshold to decrease/increase the chances of detecting a target. A critical *cutoff* value, *L(x$_0$)*, results in optimal performance with respect to the stated decision goal whenever a decision rule is based upon the likelihood ratio (23). For continuous distributions, consider the case for which *P(S|sn)* and *P(S|n)* are determined by the corresponding areas in the upper portions of two distributions:

The coordinates on the ROC curve are

$$P(S\,|\,sn)=\int_{x_0}^{\infty} f(x\,|\,sn)dx \quad \text{and} \quad P(S\,|\,n)=\int_{x_0}^{\infty} f(x\,|\,n)dx\,. \tag{2.20}$$

Differentiating both expressions with respect to the lower limit *x$_0$*

$$\frac{dP(S\,|\,sn)}{dx_0}=-f(x_0\,|\,sn) \quad \text{and} \quad \frac{dP(S\,|\,n)}{dx_0}=-f(x_0\,|\,n)\,. \tag{2.21}$$

Then, using the chain rule, the ratio can be formed as

$$\left.\frac{dP(S\,|\,sn)}{dP(S\,|\,n)}\right|_{x_0}=\frac{f(x_0\,|\,sn)}{f(x_0\,|\,n)}=L(x_0) \tag{2.22}$$

at the cutoff point, *x$_0$*. Thus, the slope on the ROC curve at *x$_0$* is equal to the likelihood ratio of the cutoff *L(x$_0$)* (23). The decision rule then divides the *L(x)* axis into *L(x) > L(x$_0$)* and *L(x) ≤ L(x$_0$)* intervals (23). As a result, the *x*-axis is divided in similar fashion into regions of acceptance (target) and rejection (non-target), as seen in Figures 2.10 and 2.12 (23). Since each cutoff, or decision threshold value, represents a diagnostic system's performance at that particular level, a 2x2 confusion matrix that summarizes the system's performance may be generated at each such point (23). In terms of *L(x)*, an

ROC curve is a function which represents the possible set of 2x2 matrices that result when the cutoff, $L(x_0)$, is varied continuously from its largest possible value to its smallest possible value (23). In terms of x, an ROC summarizes a possible set of 2x2 confusion matrices, limited by the two probability distributions selected, that results when disjoint intervals of the x-axis are successively added to the *region of acceptance*, starting with the empty set (a threshold value where no observed signal from either distribution is classified) and ending with the entire x-axis (23). In other words, the cutoff value, x_0, generates a ROC curve as the value is continually decreased across all possible values of x beginning with a very high value for x_0, as seen in Figure 2.11.

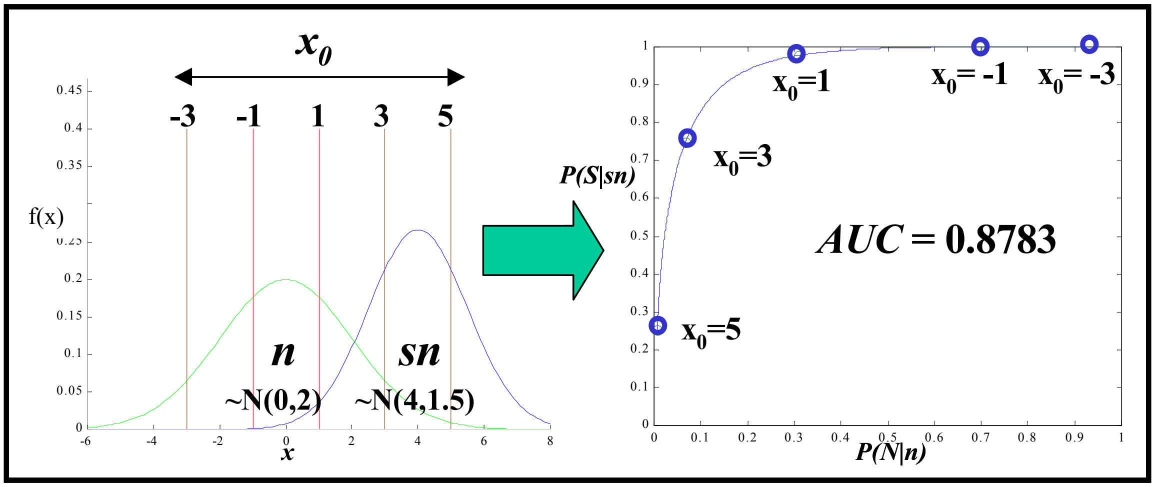


Figure 2.11 Sample N-N ROC Curve Generation.

A continuous, N-N ROC curve, $D(\Delta m,s)$, may be completely described by two parameters (as demonstrated in Figure 2.9): the distance between the means of the two normal distributions ($\Delta m = \mu_{sn} - \mu_n$) and the ratio of the target population's standard deviation to the non-target's standard deviation ($s = \sigma_{sn}/\sigma_n$). Figure 2.12 illustrates the two-parameter description system for N-N ROC curves. The left side subplots of Figure 2.12 illustrate the Gaussian distributions that represent *sn* and *n*. The difference in the means (Δm) remains constant at 0.5, while the standard deviation ratio (*s*) varies. The markers on the distribution plots indicate where the probability density of either hypothesis is the same. At these points, the slope of the ROC curves (right side subplots) at the associated point is unity, indicated by a similar marker on the ROC curve. Table 2.3 lists some of the salient features of ROC curves based on pdfs.

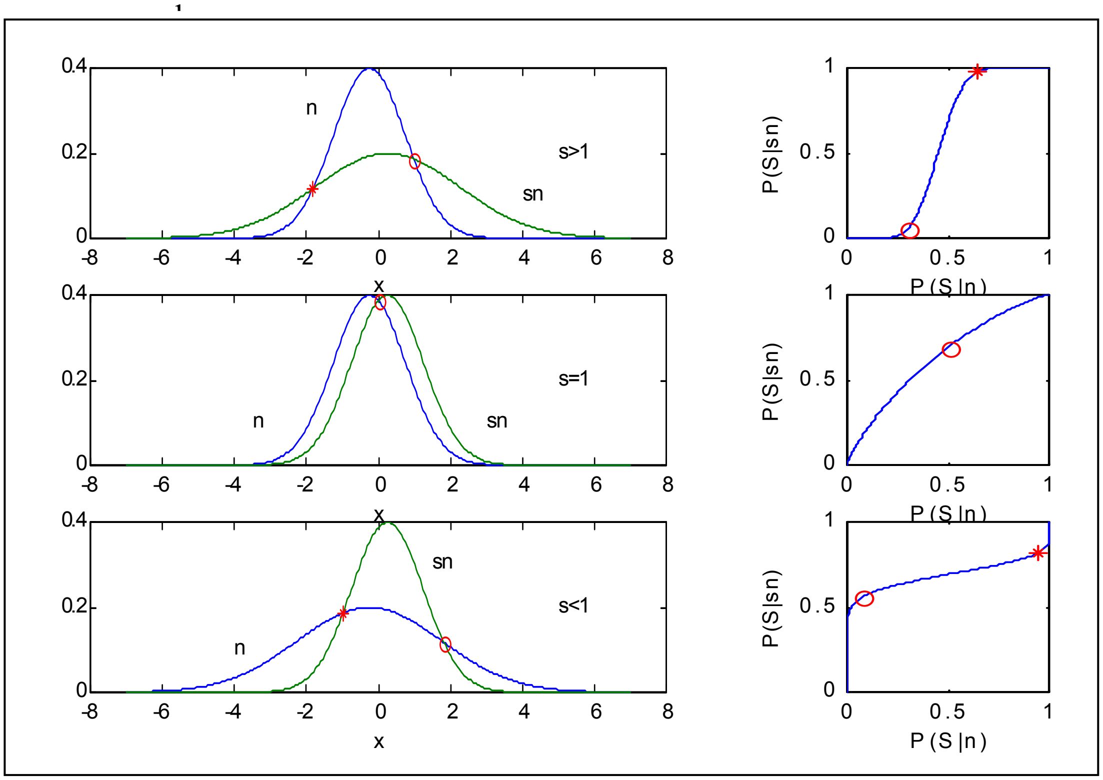


Figure 2.12 Expected ROC Curves for the Unequal-Variance Gaussian Case (26).

Table 2.3 Salient Features of 2AFC ROC Curves.

1	An ROC is based upon a pair of pdfs, one conditional upon the event *sn*, the other conditional upon the event *n*.
2	Under the assumption that the mean of the *sn* distribution is greater than that of the *n* distribution, an ROC starts at (0,0), ends at (1,1), and must be nondecreasing as long as the observer decreases the value of the cutoff point, x_0.
3	Assuming a pure decision rule is used with respect to each value of *x*, the slope of a ROC curve at a given point is equal to the likelihood ratio associated with the cutoff on the *x*-axis that generates the given point.
4	For each x_0, two disjoint, exhaustive intervals result, which generate a 2x2 matrix that corresponds to the false alarm rate (p_{FA}) and the hit rate (p_D).
5	The area under the curve, *AUC*, is equal to the percentage correct of a diagnostic system for a 2AFC task.

The ROC curve provides useful metrics and properties for the purposes of performance measurement. In the case of ATR, the ROC curve of a given CS is the graph of the probability of detection (p_D) versus the probability of false alarm (p_{FA}). This curve summarizes the possible performances of a signal detection system faced with the task of detecting a target in the presence of clutter (7). Thus, as the cutoff value is varied, the ROC curve illustrates the relationship between an ATR CS's correct target classification rate and its incorrect non-target classification rate (7). Closely related to this property is the area under the ROC curve (*AUC*) metric, which is simply the percentage correct for the 2AFC task and provides a simple index of *signal detectability* (26). This principle holds under the following assumptions: (1) that the decision in the 2AFC task is unbiased and (2) that the observations (ROIs, in the ATR CS context) are treated as statistically independent. The analytical principles have been applied to a wide number of ordered, monotonic ROC families, which may differ in the type of probability distribution used to generate the signal and clutter data (26).

Consider a two-class example problem where *n* represents the non-target class and *sn* represents the target class and with a single variable, or score, $x \in \Re$ as depicted in Figure 2.10. Let *X* be a real-valued random variable and let $p(x)$ be its pdf. Thus, $p(x|n)$ is the conditional pdf representing the distribution of non-target objects while the target pdf, $p(x|sn)$, represents the distribution of target objects. Since the choice of scale for the *x*-axis is arbitrary and is easily transformed, let higher values of *x* equate to stronger indications of target, while lower values of *x* equate to stronger indications of non-targets. The decision threshold value x_0 then divides the *x*-axis into two disjoint intervals. An observed score, *x*, found in the interval $(-\infty, x_0)$ is classified as a non-target while an observed score lying in the interval (x_0, ∞) is classified as a target. The given decision threshold boundary, $x = x_0$, then partitions the feature axis into two regions, target and non-target, resulting in two types of errors:

Type 1 Error (α): Misclassifying an actual non-target object as a target object, or False Positive (FP).

Type 2 Error (β): Misclassifying an actual target object as a non-target object, or False Negative (FN).

In Figure 2.12, the shaded portions on either side of the decision threshold line, labeled α and β, indicate the two types of error associated with the given cutoff value. This example offers a problem with a low degree of complexity. Target and non-target distributions that are nearly indistinguishable, as well as multiple class features, can easily complicate this type of performance measure (6A). However, a transformation can always be made to a simple one-dimensional space X, where X is the real-valued random variable representing the strength of conviction for the non-target (6A). Therefore, the conditional probabilities, P_{FP} and P_{FN}, corresponding to the two types of errors described above can be defined as:

$$\alpha = P_{FP}(x_0) = P(\{x > x_0 \mid n\}) = P(S \mid n) \qquad (2.23)$$

$$\beta = P_{FN}(x_0) = P(\{x < x_0 \mid sn\}) = P(N \mid sn). \qquad (2.24)$$

Associated with these two probabilities are their complementary probabilities of correct classification, where TP stands for True Positive and TN stands for True Negative:

$$P_{TN}(x_0) = P(\{x < x_0 \mid n\}) = P(N \mid n) \qquad (2.25)$$

$$P_{TP}(x_0) = P(\{x > x_0 \mid sn\}) = P(S \mid sn). \qquad (2.26)$$

The interrelationships among these probabilities and the various terminologies used in ATR, statistics, and medicine to describe them are shown in Figure 2.3. Due to these interrelationships, a collection of probability pairs is all that needs to be reported to describe the performance of an ATR CS for a particular decision threshold value, x_0 (6A). The probabilities correspond to specificity, the power to correctly declare true negatives, and sensitivity, the power to correctly declare true positives, of the CS.

The major strength of ROC curves in CS evaluation is that they do not simply report the system's performance in terms of a target detection *batting average* for a specific decision threshold. ROC curves enable performance reporting in terms of a pair of related indices (detection probability, false alarm probability) for varying thresholds, as seen in Figure 2.11 (6A). ROC curves provide a means for characterizing and quantitatively comparing CS designs (6A). In other words, two ATR CSs can be compared over multiple decision thresholds and over the same feature space by a single ROC curve for each CS, usually via the *AUC* metric. This will be examined in depth in the section dealing with ROC curve comparison metrics.

Variants of the ROC curve include: the frequency ROC (FROC), the expected utility ROC (EUROC), the localization ROC (LROC), the response analysis characteristic (RAC) curve, and the operating curve (29,36,66). The FROC and EUROC, occasionally used in the field of biomedical analysis, merely change the parameter used when generating the ROC curve (36). The LROC, or joint ROC, is a version of the ordinary ROC curve that allows the CS to choose a confidence rating and one of *m* alternatives in its classification decisions (61). The RAC curve depicts the inverse of both the probability of false alarm, $P(S|n)$, and the probability of detection, $P(S|sn)$ (61). Thus, the RAC curve is a plot with $P(n|S)$ and $P(sn|S)$ as the *x* and *y* axes (61). The RAC is practically useless in that the curves generated takes forms that are not easily indexed and the range of the curve depends upon the prior probabilities (61). The operating curve uses the misclassification of known targets as the *y*-axis ($1-p_D$) and retains the false negative axis of the typical ROC curve (30). The result is a curve that demonstrates the power of misclassification by the ATR CS (30). Like the ROC curve, the area under the curve of an operating curve is the preferred performance index in assessment. A near perfect ATR CS generates an area under the curve close to zero (30). Figure 2.13 depicts an operating curve.

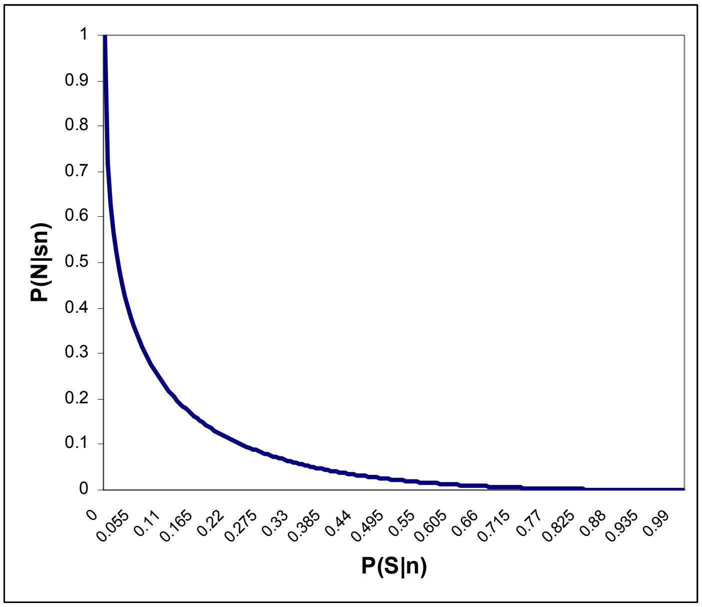


Figure 2.13 Operating Curve Derived from 2AFC Task.

Current ROC analysis research, particularly in the field of biostatistics, seems to focus on several different areas of concentration. First, there is considerable work accomplished on the different ways to calculate the *AUC* for a given ROC curve (17,36,43,66). The estimation of *AUC* is very important, especially when comparing two or more competing CSs. Thus, superior and robust methods of producing an accurate *AUC* value are desired. Another area of ongoing research is that of ROC curve meta-analysis, which is the estimation of the true ROC curve of a given diagnostic system through ROC analysis across many studies or trials (38,44,53,70). Capturing an estimate of the true, overall ROC curve for a given CS is the goal. Using data from several different studies, tests, and trials introduces problems in the way that the data is meshed or weighted in order to produce the best ROC *AUC* estimate. Finally, a new approach to ROC curve analysis is that of three-way ROC curves (21). In this case, the diagnostic system yields not only a diseased/non-diseased (target/non-target) decision, but also a third outcome, such as an *undecided* decision (21,47). This third decision alternative relates to the *declaration* question in a typical ATR. Instead of using the *AUC* metric, a new metric, the volume under the surface (*VUS*), is used. This new approach realizes new challenges in the comparison of multiple diagnostic systems.

2.3 Automatic Target Recognition Performance Measure Comparison

2.3.1 Background.

There is considerable theory and literature associated with the concept of performance measure comparison. This section is broken into two sections that correspond to the two types of comparison: the use of visual techniques and the use of mathematical techniques to compare multiple systems via their performance measures. The first section lists the various graphical techniques used in the comparison of ATR CSs. The second section highlights the various mathematical comparison techniques used in the comparison of ATR CSs.

2.3.2 Visual Techniques.

The strength of these visual techniques is the simplicity with which they can be used and their inherent appeal to the human ability to visually compare objects via size, color, and shape. These techniques can typically frame the comparison in a way that is easily understandable to a decision maker and can often associate multiple dimensions, large amounts of data, and relationships between several systems in a single plot.

2.3.2.1 Visualization Guidelines.

Multidimensional variables are often difficult for humans to compare, especially when listed in a numerical fashion, such as a table. Graphs and plots offer an analytical window into the trends, oddities, and pertinent features of multi-dimensional data.

With the several tasks and concepts to remember when approaching an abstract scientific visualization problem, visualization specialists often implement an algorithm of engineering guidelines to follow (49). The typical algorithm highlights several effective means of visually displaying multivariate data and associated problem solutions. One such algorithm uses eight engineering design guidelines to lead an analyst through the process of visually describing a problem and its solution. The guidelines are generalized in order to accommodate the various problem types encountered by analysts. The authors admit that some of these guidelines may not apply to a given problem, but they expect the algorithm to be useful for most tasks and applications (49). Table 2.4 lists the eight guidelines for visualizing multivariate data and problem solutions.

Table 2.4 Eight Visualization Engineering Guidelines (49).

	Engineering Guideline	Description
1	Task-specific	Plots and graphs answer the questions of interest
2	Reduced Representation	Fit data on one screen, if possible
3	Data Encoding	Use glyphs, markers, & colors to represent data
4	Filtering	Reduce amount of data shown through filter rules
5	Drill Down	Visualize data not currently onscreen
6	Multiple Linked Views	Update plot changes on all plots at once
7	User Interface	Allow user to manipulate plots/data
8	Animation and Motion	Animate temporal or physical aspects of data

The first engineering guideline is to keep the visualization task-specific. The most important thought in visually depicting a problem and its solution should be to keep the user's needs in mind. In other words, being task-specific in scientific visualization ensures that the graphs and plots are illustrating what the user wants to know in a way that he or she may understand. If the graphs do not answer the user's questions, then the graphs and plots are irrelevant. Therefore, this guideline is of utmost importance.

The concept of utilizing reduced representation is the next guideline to consider. This concept refers to illustrating the entire data set or results on a single screen. Doing so avoids a user having to flip through several different *pages* of results and allows the user to compare results or different areas of data easily. Accomplishing this task is not always easy. Care must be given to the type and position of glyphs used to represent the data. Also, showing all of the data onscreen should not be something that overwhelms the user; it should just be a quick, efficient way to compare trends and dimensions within the data.

Data encoding, the third engineering guideline, refers to employing the physical attributes of the glyphs effectively in the graphs and plots. Size, shape, color, and position are a few of the attributes that can be used to annotate information within the data.

Another concept worthy of consideration is that of filtering. Filtering allows the user to toggle between certain aspects of the data. For instance, if the data set has several features used to classify an observation, the user could select to view only a few of the features at one time. This creates a manageable, uncluttered view for the user. This technique can be very useful for very large data sets with many variables.

The drill down technique allows a user to find out information about a feature or observation that is not currently onscreen. For instance, a user could drag a mouse pointer over a given data point represented by a glyph in a plot, and a pop-up window would display the numerical location of the point. This technique can be very effective in sensitivity analysis; trying to determine which data points have the most effect on the solution, and in determining outliers in a data set.

The sixth guideline, incorporating multiple linked views, suggests that, when using multiple views of the same or similar graph and plots, changes to one plot should carry over to the subsequent plots. This concept creates harmony in the viewing of complex data and may uncover trends that may have remained undetected. The technique also saves the user time by automatically changing several plots rather then the user manually changing each plot of interest.

The seventh guideline shows concern for the one actually visualizing the data by calling for user interface of the plot or graph. Plots and graphs of large or complex data sets should incorporate the ability to be directly manipulated by the user. Some of the features may include plot axes changes, plot zooming features, rotateable images, and data set selection. The authors also suggest that this process must be executed quickly and that sluggish imaging performance greatly reduces the effectiveness of scientific visualization by disenchanting the user.

Use of animation and motion is the premise behind the final data visualization guideline. This guideline refers primarily to data sets that have a temporal aspect. Motion and animation can be key in determining changes in the data. One difficulty with this technique is the computational power afforded by the visualization platform.

2.3.2.2 Confusion Matrices.

Confusion matrices are an efficient and orderly way of presenting the pertinent performance information for a given ATR CS. The matrix structure allows for quick comparison between the numeric performance measures of several CSs. The performance measure probabilities are typically the items of interest in a given confusion matrix. Figure 2.14 illustrates the use of confusion matrices in the comparison between two different ATR CSs, systems A and B. CS B seems to perform better as a target classifier overall, but does not do as good a job on clutter when compared to CS A. CS A's major deficiency is the tendency to report actual targets as clutter (false negatives), seen in that it misclassified nearly 60% of the Scud MRLSs as clutter. CS B's major weaknesses are its tendency to misclassify detected T72s and to declare actual clutter as a target (false positives).

Classification System A

Classified As (Reported)

		T72	M1	Scud	Non-TGT
	T72	12 (86%)	2 (14%)	0 (0%)	0 (0%)
Actual	M1	2 (14%)	8 (57%)	3 (22%)	1 (7%)
(Truth)	Scud	0 (0%)	0 (0%)	7 (41%)	10 (59%)
	Non-TGT	0 (0%)	1 (3%)	5 (12%)	35 (85%)

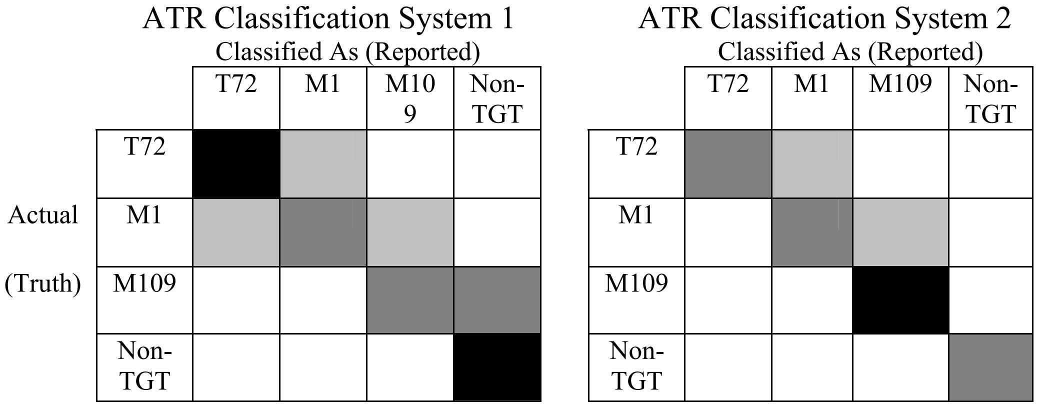
Classification System B

Classified As (Reported)

	T72	M1	Scud	Non-TGT
T72	10 (72%)	2 (14%)	1 (7%)	1 (7%)
M1	0 (0%)	10 (72%)	4 (28%)	0 (0%)
Scud	0 (0%)	1 (6%)	15 (88%)	1 (6%)
Non-TGT	3 (7%)	2 (5%)	5 (12%)	31 (76%)

Figure 2.14 Example Composite Confusion Matrices.

While a typical confusion matrix contains the numerical performance description of a given CS, the confusion matrix can be transformed into a matrix of shaded blocks that correspond to the numeric values of the original matrix (7). This technique enables an evaluator to identify the strengths and inadequacies of a particular CS and differences between multiple CSs through the use of color. The darker the shading of a square in the grid indicates that the classifier associated (classified) a detected target to the given row (truth) more frequently (7). A near perfect classifier produces a confusion matrix with a very dark right hand diagonal and very pale entries elsewhere (7). Figure 2.15 illustrates the use of shading in confusion matrices for the purposes of comparing the same two ATR CSs. Viewing the matrices indicates that the CS B does a better job at correctly classifying M1s and Scud MRLSs, which is the same conclusion drawn from the composite confusion matrix comparison in Figure 2.14. The darker diagonal of CS B indicates it is closer to a near-perfect classifier than the CS A. However, depending on the desired goal, CS A could be considered better. For instance, if the objective is to find the CS that most correctly classifies T72 MBTs, then CS A, which correctly classified 86% of the T72s, could be considered the better system.



ATR Classification System 1

		Classified As (Reported)			
		T72	M1	M109	Non-TGT
	T72	■	░		
Actual	M1	░	▓	░	
(Truth)	M109			▓	▓
	Non-TGT				■

ATR Classification System 2

	Classified As (Reported)			
	T72	M1	M109	Non-TGT
T72	▓	░		
M1		▓	░	
M109			■	
Non-TGT				▓

Figure 2.15 Typical Gray Level Confusion Matrices.

Though it allows quick comparison between the performances of two systems, one disadvantage of the gray level confusion matrix is the inability to distinguish between slight differences of color. For instance, it is difficult for the human eye to detect a difference between a 75% gray level and an 80% gray level. This disadvantage requires a better method of visual comparison when dealing with CSs that are very similar in performance. The previous example offered two classification systems with stark contrasts. Thus, the advantages of the technique were readily apparent.

2.3.2.3 Error-Reject Curves.

Doubt reports are a method of allowing a pattern recognition classifier to report confidence in its ability to correctly classify an object (7). Due to the assumed distribution of targets and non-targets, certain objects detected by the CS are difficult to classify, i.e. the given score of a detected object lies close to the decision threshold between the target and non-target distributions. These objects are then rejected until further measurements that lead to a more definite classification can be made or these objects may be transferred to a second stage classifying system designed to deal with objects that require finer classification precision (7). A loss function can be defined as the loss incurred by making decision *l* if the true class is *k* (out of *K* classes) (7). If every misclassification is equally serious, then the loss function is given by:

$$L(k,l)=\begin{cases}0 & if\ l=k\ (correct\ classification)\\ d & if\ l=D\ (classification\ in\ doubt)\\ 1 & if\ l\neq k\ and\ l\in\{1,...K\}\,(incorrect\ classification)\end{cases}\qquad(2.27)$$

where $k = 1,\ldots,K$ and $l\in\{1,\ldots,K\}$ is a reasonable choice. The total risk for the optimal decision rule is called the Bayes' risk (R) and is defined by

$$R = p_{MC} + d \cdot p_d \qquad (2.28)$$

where p_{MC} is the probability of misclassification, p_d is the probability of doubt, and d is the rejection threshold, or the cost of being in doubt (7). Error-reject curves are the plot of p_{MC} versus p_d (7). These curves are particularly useful is describing the relationship between making a classification error and the doubt associated with a classification decision. Since the slope of the error-reject tradeoff curve is the value of the rejection threshold, the tradeoff is most effective for low levels of rejection and becomes less effective for high levels of rejection when the error rate is very low (7). Figure 2.16 provides an example of an error-reject curve.

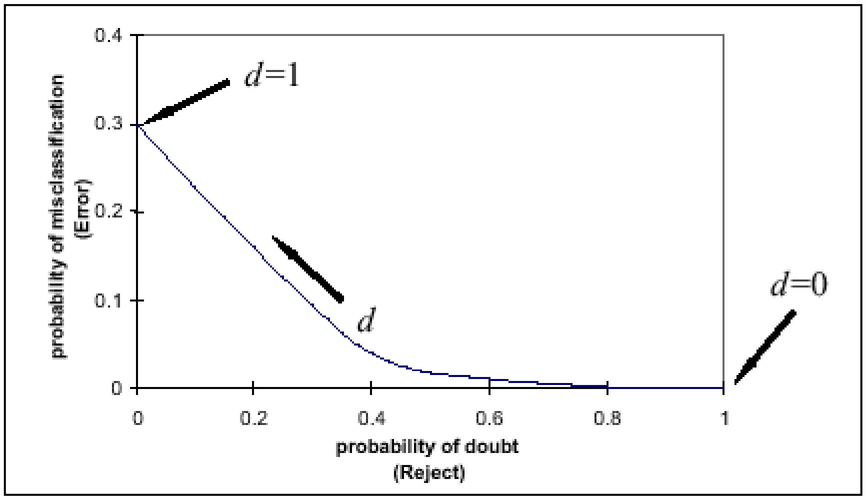


Figure 2.16 Error-Reject Tradeoff Curve.

2.3.2.4 Error Histograms.

For classifiers with several outputs or in situations where the size of the errors is more important than their type, an error histogram provides another quick method for visualizing the distribution of errors. An error histogram shows the count of the frequency with which a classification error falls within a set of bandwidths, i.e. within a certain range of error sizes. These bandwidths or error sizes are the ranges of possible differences between the actual target class and the predicted class for each exemplar. For a classification probability score from zero to one, these bands must be split into a small set of bins. This error binning technique contrasts the setting of class thresholds used to classify the exemplars and generate the confusion matrix. For a simple two-class confusion matrix, if the predicted classification score for a particular exemplar exceeds some preset threshold, then that exemplar is classified as class 2. For the error histogram, the difference between a given exemplar's predicted classification probability and each target output class probability is used. A healthy classifier will show a peak at zero, quickly falling off as the number of errors of greater magnitude diminishes. For a data set with normally distributed noise, the error histogram should have the appearance of a normal distribution. Figure 2.17 shows an example of the error histogram of a healthy classifier. The error histograms of competing classifiers can be examined to identify differences in the performance of multiple CSs or deficiencies in a single CS.

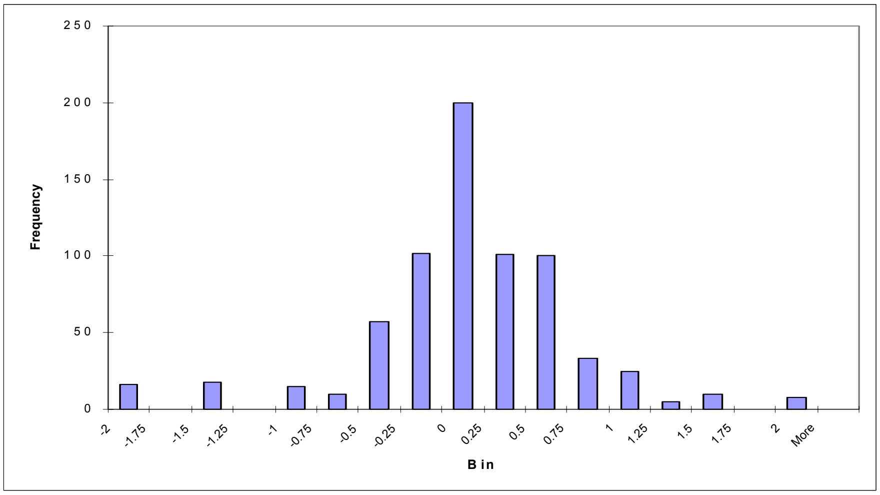


Figure 2.17 Error Histogram Example.

2.3.2.5 Classification Trees.

In biomedical decision-making scenarios, classification trees are used to aid the diagnosis between diseased and healthy patients (5). Often this technique combines information from one or more diagnostic tests with patient characteristics to better identify patients with the disease of interest (5). Another advantage of this technique is that the leaves of a classification tree provide enough data for ROC curve generation (5). This technique could be applied to the various questions raised in DUSD-BM evaluation. For instance, the selection of a superior ATR CS by allowing questions to systematically rank the performance of a CS or "weed out" an inferior CS that does not meet AFRL/SN performance requirements. The following simplistic scenario uses a classification tree to identify a superior CS.

Suppose that for a combat identification (CID) scenario, AFRL/SN requires a p_{ID} of 0.87 and a p_{FA} of 0.2. Since each CS can presumably achieve these values dependent upon their detection threshold value setting, the best way to compare the CSs is to compare the other metric at a specific performance measure criterion. For instance, if the most important performance measure for a CID scenario is p_{FA} then the selection should be based on the superior p_{ID} at the required p_{FA} level. The objective of the decision is to determine which two ATR CSs should be chosen for further competition and improvement. The CSs competing for contract selection, along with their performance measures, are listed in Table 2.5. Notice that the performance measures of each CS indicate that no dominant CS exists for consideration. Therefore, the decision maker must weight the preferences. Pertinent questions from the decision maker's perspective, and implied by the rules of the program, are ranked by importance and used in the classification tree. The questions for this simplistic scenario are given in Table 2.6. The first two questions attempt to divide the CSs on the CID requirements. The final question separates the CSs on overall performance. Figure 2.18 provides a depiction of the classification tree and the results. The ATR CSs are rank-ordered from right to left. Thus, the CS performance ranking in descending order is 4,2,1,3.

Table 2.5 ATR CS Performance Measures for Competitive Selection Scenario.

ATR	p_{ID} (Rank)	p_{FA} (Rank)	*AUC* (Rank)
1	0.82 (3)	0.3 (4)	0.878 (4)
2	0.87 (2)	0.18 (1)	0.823 (3)
3	0.77 (4)	0.28 (3)	0.902 (1)
4	0.93 (1)	0.25 (2)	0.899 (2)

Table 2.6 Rank-Ordered Questions for Competitive Selection Scenario.

Number	Question	Metric
1	Does the CS have a $p_{ID} \geq 0.80$ @ the CID p_{FA} requirement?	p_{ID}
2	Does the CS have a $p_{FA} \leq 0.25$ @ the CID p_{ID} requirement?	p_{FA}
3	Does the ATR CS have a higher ROC *AUC* value than its competitor at this classification tree level?	*AUC*

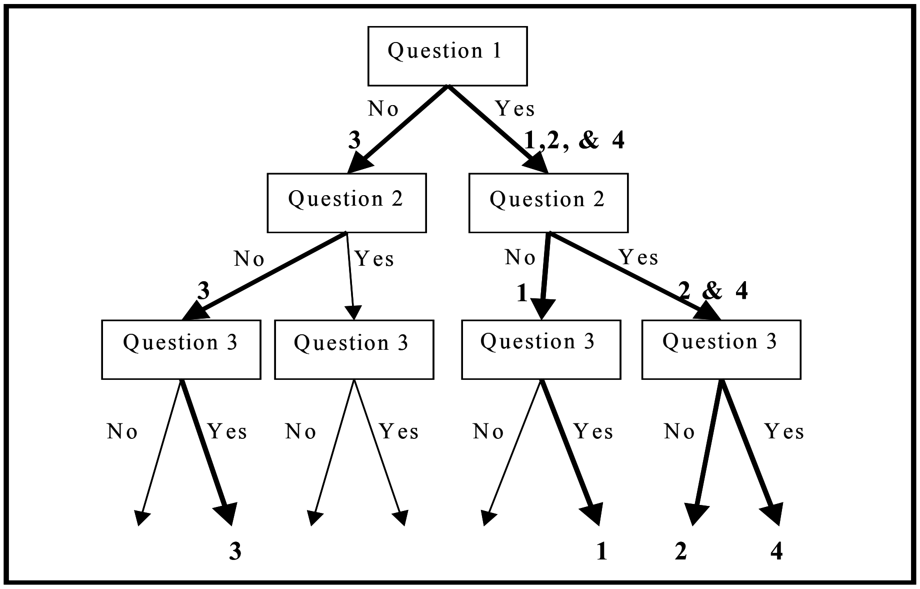


Figure 2.18 Classification Tree for Competing ATR CS Selection Scenario.

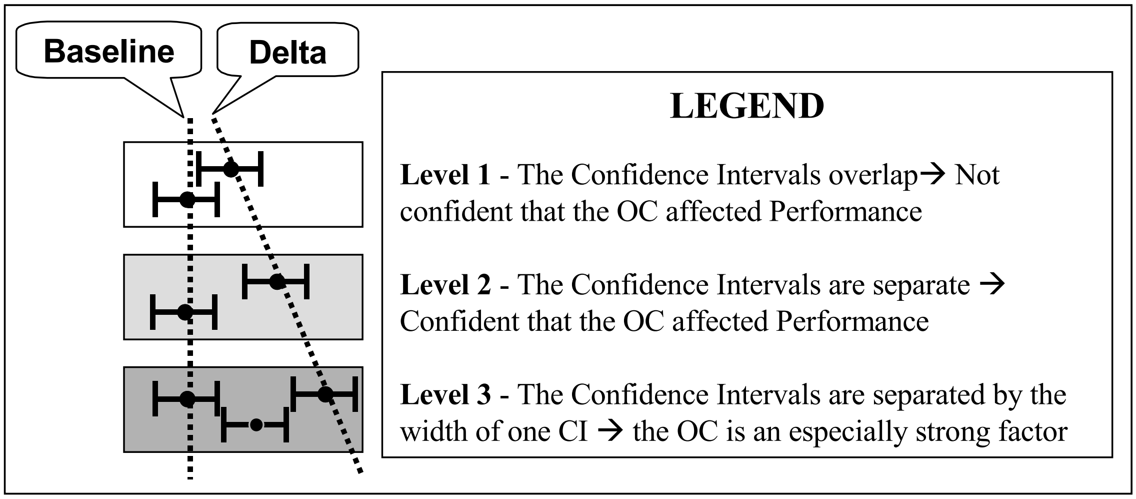
Notice that in this scenario, the classification tree used is a slightly modified version of that used in biomedical decision-making. The typical classification tree would implement only the right–hand-side of the tree depicted in Figure 2.18. The above tree illustrates the ranking system for all CSs, but could have ended after the second question since two CSs are found superior to the others. Finally, notice that the final question is a one that does not guarantee a single outcome. The answer to this question could result in multiple “no” responses. A more complex decision could require further questions to rank and separate CSs below the third classification level.

2.3.3 Statistical Techniques.

Statistical techniques are particularly useful in performance measure comparison because they typically provide not only a structured method of comparison but also a level of certainty in the comparisons made. The major drawbacks of using statistical methods are the reliance on the assumptions made about the data and the lack of confidence associated with a given comparison due to the lack of data. The methods discussed here include statistical representations that are used to make inferences about comparisons (hypothesis testing), indicate differences between several competing systems (ROC curve performance measures and the multinomial selection procedure), evaluate multi-goal decision-making (linear goal programming), and employ decision analysis (DA) to decide amongst several competing classifiers.

2.3.3.1 Confidence Intervals.

The AFRL COMPASE Center uses confidence intervals to compare sensitivity of performance of several ATRs. During ATR testing, a known target is presented in its most basic configuration. This configuration represents the data on which the ATR CS was initially trained. For example, if a T72 tank were used as the baseline target, the turret would be positioned straight ahead. No external devices would be attached to the vehicle, and the tank would be positioned in the field clear of clutter and revetments. The probabilistic performance measures, such as p_D, for each individual ATR CS are computed for the baseline target. The performance measures collected on each ATR CS serve as a benchmarking measure for the ATR’s performance at a nominal setting. In other words, an ATR CS's detection and classification performance on the baseline target is considered to be in the optimal expected performance region for the CS. Changes to the OCs are expected to degrade system performance, or, at best, remain unchanged. Performance measures from the remainder of the test, where the OCs are varied, are calculated and subtracted from the baseline performance measures to create a *deviational* (delta) performance measure. Confidence intervals are constructed around the deviational and baseline performance measures. The confidence intervals are then used to determine performance deviations from the benchmark performance measures. Thus, each ATR CS is compared to its own "optimal" performance, and the deviational results are used to compare ATR CS's across the board. This technique does not use raw probabilistic performance data, such as p_D, to compare various CSs. Rather, a difference between baseline and deviated performance figure is used. Figure 2.19 demonstrates how the confidence interval comparison results are presented. Table 2.7 provides a sample chart of how the results are used for CS comparison. In Table 2.7, all deviations are considered degradations. In other words, the inclusion of OC variation introduced a degraded ATR CS performance.



LEGEND

Level 1 - The Confidence Intervals overlap→ Not confident that the OC affected Performance

Level 2 - The Confidence Intervals are separate → Confident that the OC affected Performance

Level 3 - The Confidence Intervals are separated by the width of one CI → the OC is an especially strong factor

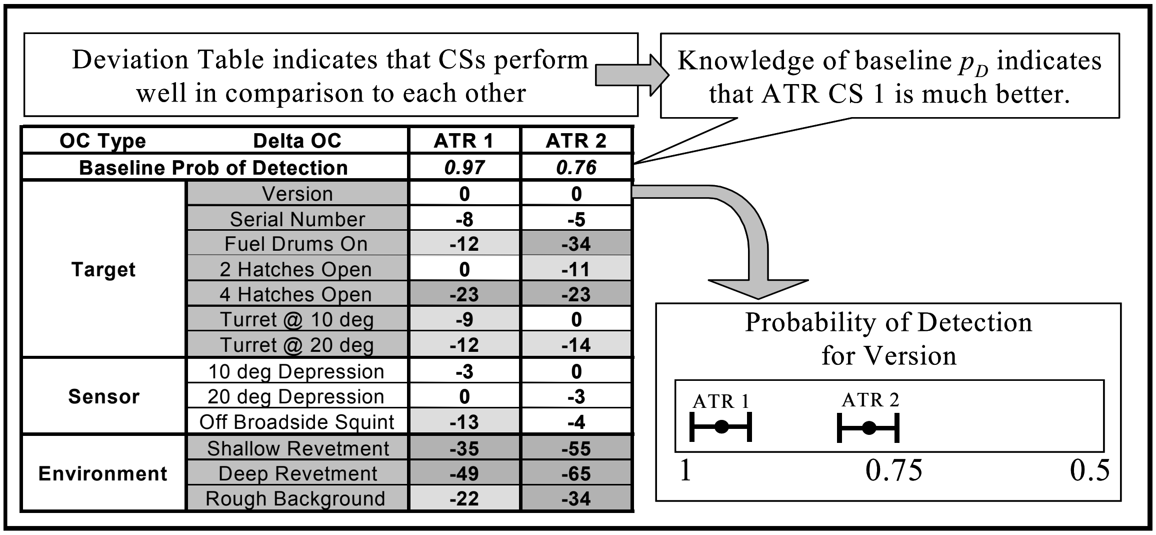
Figure 2.19 Demonstration of Confidence Interval Use in ATR Comparison (56).

Table 2.7 Example ATR CS Comparison Using Confidence Intervals (56).

OC Type	Delta OC	ATR 1	ATR 2	ATR 3
Target	Version	**0**	**0**	**-1**
	Serial Number	**-8**	**-5**	**-2**
	Fuel Drums On	**-12**	**-34**	**0**
	2 Hatches Open	**0**	**-11**	**-12**
	4 Hatches Open	**-23**	**-23**	**-16**
	Turret @ 10 deg	**-9**	**0**	**-1**
	Turret @ 20 deg	**-12**	**-14**	**-2**
Sensor	10 deg Depression	**-3**	**0**	**-15**
	20 deg Depression	**0**	**-3**	**-22**
	Off Broadside Squint	**-13**	**-4**	**0**
Environment	Shallow Revetment	**-35**	**-55**	**-27**
	Deep Revetment	**-49**	**-65**	**-67**
	Rough Background	**-22**	**-34**	**-44**

A limitation of this technique is that the differences in the benchmark performance for each ATR CS are not given. Therefore, a CS may perform poorly on the baseline target while every other CS performs well. The poor performer may have similar deviational scores, but the deviation values are from a lower performance. Thus, the CS should not be considered for comparison between the other CSs. Figure 2.20 illustrates this problem. However, performance of CSs against the baseline is generally near perfect since the target is in the configuration upon which the ATR CS was trained. Thus, all CSs are relatively equal when detecting and classifying the baseline target.

Higdon proposed improvements to the ATR evaluation process that would incorporate a factorial design rather than a one-at-a-time test design currently being used by the AFRL COMPASE Center (35). Results from simulated data demonstrated that such a design offers more efficient identification of significant relationships between features, or OCs, and more accurate confidence interval estimation for performance measures (35).



Deviation Table indicates that CSs perform well in comparison to each other

Knowledge of baseline p_D indicates that ATR CS 1 is much better.

OC Type	Delta OC	ATR 1	ATR 2
Baseline Prob of Detection		*0.97*	*0.76*
Target	Version	0	0
	Serial Number	-8	-5
	Fuel Drums On	-12	-34
	2 Hatches Open	0	-11
	4 Hatches Open	-23	-23
	Turret @ 10 deg	-9	0
	Turret @ 20 deg	-12	-14
Sensor	10 deg Depression	-3	0
	20 deg Depression	0	-3
	Off Broadside Squint	-13	-4
Environment	Shallow Revetment	-35	-55
	Deep Revetment	-49	-65
	Rough Background	-22	-34

Probability of Detection for Version

Figure 2.20 Graphical Depiction of Limitations when Using Confidence Intervals to Compare ATR CSs.

2.3.3.2 Hypothesis Testing.

Hypothesis testing can be used to determine if there is a statistically significant difference between the performance measures of two or more systems (41). When comparing two CSs, one can decide in advance the number of trials for testing each system (non-sequential) or one can have the testing procedure decide *on the fly* (sequential) (7).

When using non-sequential testing to compare two CSs, one can compare the confidence intervals for some performance measure p for both systems, or the confidence interval for the difference in performance between the two systems can be computed. For large *n* (*n*>30) and assuming equal sample sizes ($n = n_1 = n_2$), the difference interval can be calculated as:

$$(\hat{p}_2 - \hat{p}_1) \pm Z_{1-\alpha}\sqrt{\frac{\hat{p}_1(1-\hat{p}_1)+\hat{p}_2(1-\hat{p}_2)}{n}} \tag{2.29}$$

with the associated test statistic:

$$Z_0 = \frac{\hat{p}_2 - \hat{p}_1}{\sqrt{\frac{\hat{p}_1(1-\hat{p}_1)+\hat{p}_2(1-\hat{p}_2)}{n}}}. \tag{2.30}$$

Then the following hypothesis may be made:

$$H_0 : p_1 \geq p_2 \tag{2.31}$$

$$H_A : p_1 < p_2. \tag{2.32}$$

This technique is useful for comparing two CSs, but requires large sample sizes, especially for distinguishing between small differences in the performance measures of the two systems. As before, the assumption that the probability *p* does not vary from trial to trial makes this technique inappropriate for use in some ATR applications. However, Wald’s non-sequential testing procedure is a cleverly simple method for probability comparison, which allows for the variation of probabilities from trial to trial. In comparing two ATR CSs, there are two possible outcomes, represented by

$$\eta = \begin{cases} 0 & if\ t\arg et\ ROI\ incorrrectly\ classified\ as\ non-t\arg et \\ 1 & if\ t\arg et\ ROI\ correctly\ classified\ as\ t\arg et \end{cases}. \quad (2.33)$$

The results are arranged in pairs in the ordered observed, i.e. t=(η_1,η_2) where *1* and *2* correspond to the two ATR CSs. Thus, the number of observations where the first ATR CS correctly identified the ROI while the second CS did not, represented by (1,0), are denoted as t_1. The other outocme, denoted t_2, is the opposite case (0,1). Therefore, a hypothesis test may be generated using

$$H_0\text{: } p \geq 0.5 \quad (2.34)$$

$$H_A\text{: } p < 0.5 \quad (2.35)$$

where p is the probability that any ordered pair (*a,b*) is equal to (0,1) and is given by

$$p = \frac{(1-p_1)p_2}{p_1(1-p_2)+p_2(1-p_1)}. \quad (2.36)$$

The test statistic for the equivalent hypothesis tests is simply the number t_2 of observed ordered pairs (0,1). The null hypothesis, that p_1 is better than p_2, is rejected only if, $t_2 \geq T$, where the value of T, for a given level of significance α, is given by the binomial distribution with p = 0.5:

$$P(t_2 \geq T) = \sum_{t=0}^{T} \binom{t}{i} p^i (1-p)^{t-i} = 1-\alpha \quad (2.37)$$

where $t = t_1 + t_2$.

An extension of Wald's procedure can be applied in a sequential hypothesis testing approach (7). The Wald sequential test is based on the efficiencies of the two competing ATR CSs. Efficiency is defined as

$$k = \frac{p}{(1-p)} \tag{2.38}$$

such that p is the true probability of success. The relative superiority of a second CS over the first CS is measured by the ratio (*u*) of the efficiencies:

$$u = \frac{k_2}{k_1} = \frac{p_2(1-p_1)}{p_1(1-p_2)}. \tag{2.39}$$

For the test, four parameters must be set, which reflect the precision required (u_0, u_1) and the tolerated risks (α,β). Also, the test statistics and hypotheses are constructed as in Wald's non-sequential test. However, t_2 is compared to two critical values: the acceptance and rejection numbers, given by

$$\text{Acceptance number:}\quad a_t = \frac{\log\frac{\beta}{1-\alpha}}{\log u_1 - \log u_0} + (t_1 - t_2)\frac{\log\frac{1+u_1}{1+u_0}}{\log u_1 - \log u_0} \tag{2.40}$$

$$\text{Rejection number:}\quad r_t = \frac{\log\frac{1-\beta}{\alpha}}{\log u_1 - \log u_0} + (t_1 - t_2)\frac{\log\frac{1+u_1}{1+u_0}}{\log u_1 - \log u_0}. \tag{2.41}$$

If t_2 falls below the acceptance number for any value of *t*, the null hypothesis that ATR CS_1 is better than ATR CS_2 is accepted. If t_2 exceeds the value for the rejection number, the null hypothesis is rejected and the conclusion is that ATR CS_2 is better than ATR CS_1. If t_2 remains between these bounds, testing is continued.

The Wald sequential test procedure has been applied in comparing the p_{ID} performance measure for different configurations of the MSTAR system using actual data and in a four system comparison, with an embedded Wald sequential test methodology in a multiple sequential rejective Bonferroni procedure, using simulated data (15). The results indicated an improved sample size savings advantage through the use of the Wald procedure, which is of great benefit since image data collection is very expensive (15).

2.3.3.3 ROC Curve Performance Measures.

The most commonly used index for comparing ROC curves is the area beneath the ROC curve (*AUC*) (7,8,9). This area is equivalent to the probability of success for a diagnostic system identifying both target and clutter images in a series of image pairs in which there is always a target and clutter image (27). Again, the *AUC* measure represents a convenient and simple index of target detectability. However, deficiencies in the *AUC* performance measure have pushed the search for better ways to compare ROC curves. ROC analysts in the field of biostatistics discourage the use of *AUC* when the ROC curves for classification systems overlap or are mismatched (17,43). The *AUC* measure has been shown to fail in the definition of being a true metric (7). For example, two ROC curves may have different shapes, but have the same *AUC* value. With respect to metric definition, this violates the *definiteness* property (7).

Several measures have been proposed as alternatives to the *AUC* measure. First, the area under the binormal ROC, denoted A_z, is the area under the ROC curve that is above the diagonal chance line (66). Thus, A_z ranges from 0.5 to 1.0 and provides a measure of how a diagnostic system performs in relation to chance. This measure is "less affected by the location or spread of the points" that compose the ROC curve (66). The A_z measure assumes that the target and non-target distributions may be modeled with normal pdfs. Then, the ROC curve may be plotted on a binormal graph, where the ROC curve is linear. The result is an easily calculated area under the curve that does not underestimate the area under the ROC curve by using an approximation rule. The *AUC* calculation method for empirical data is typically computed using the trapezoidal rule, or some other approximation rule, on a linear probability scale thereby underestimating the true area under a complete ROC curve. One can argue that the *AUC* measure is superior in that it makes no assumption concerning the underlying distribution of the ROC like the A_z measure. In the context of ATR evaluation, this method is of little use since no assumption is made concerning the underlying target and non-target distributions.

A proven metric, called the average metric distance, has been proposed for the comparison of multiple ROC curves associated with multiple CSs (7). Since the area under the curve may be the same for two different ROC curves, this metric does not incorporate the idea of area under the ROC curve. Rather, the metric measures the average metric distance between two ROC curves in order to estimate the difference in their respective *AUC* measurements. The average metric distance metric can be approximated to range from 0 to 1, like the *AUC* measure, where 0 implies no difference between two ROC curves and 1 implies maximum difference between two ROC curves (7). This discrete approximation is represented by:

$$AverageMetricDistance \approx \frac{\sum_{i=1}^{m} \rho_q(P^{(f)}(\theta_i), P^{(g)}(\theta_i))}{m} \tag{2.42}$$

where

$$\rho_q(\vec{x}, \vec{y}) = (|x_1 - y_1|^q + |x_2 - y_2|^q)^{\frac{1}{q}} \ , \tag{2.43}$$

and

$$\rho_\infty(\vec{x}, \vec{y}) = \max\{|x_1 - y_1|, |x_2 - y_2|\} \tag{2.44}$$

for each $1 \leq q < \infty$. For the above equations, ρ is the distance metric, q corresponds to the type of distance metric implemented, m is the number of thresholds evaluated, θ is the individual threshold value where each ROC curve is evaluated, and $P^{(f)}$ and $P^{(g)}$ represent the two ROC curve data sets. The vectors $\vec{x}$ and $\vec{y}$ correspond to the points of two ROC curves under comparison via the distance metric. The average distance metric also allows the use of any available distance metric, such as the Euclidean or Manhattan metrics, which determines the distance between two points on two separate ROC curves (7). Most importantly, each ROC curve can be compared against a known reference curve, such as the negative diagonal line. Doing so corresponds directly to a difference comparison in the AUC measurement between classifiers. Another strength of this metric lies in the fact that the distance measure is based upon the threshold placement, not the number of samples from the target and non-target distributions. Finally, all calculations of this metric are perpendicular to the negative diagonal line, as seen in Figure 2.21, which illustrates the calculation of this ROC performance metric. Note that the distance measure*, di,j,* depicted in the plot is based on the distance metric selected by the user. In the plot, this same measure is shown at the 120th threshold for each CS and is based upon the Euclidean distance measurement.

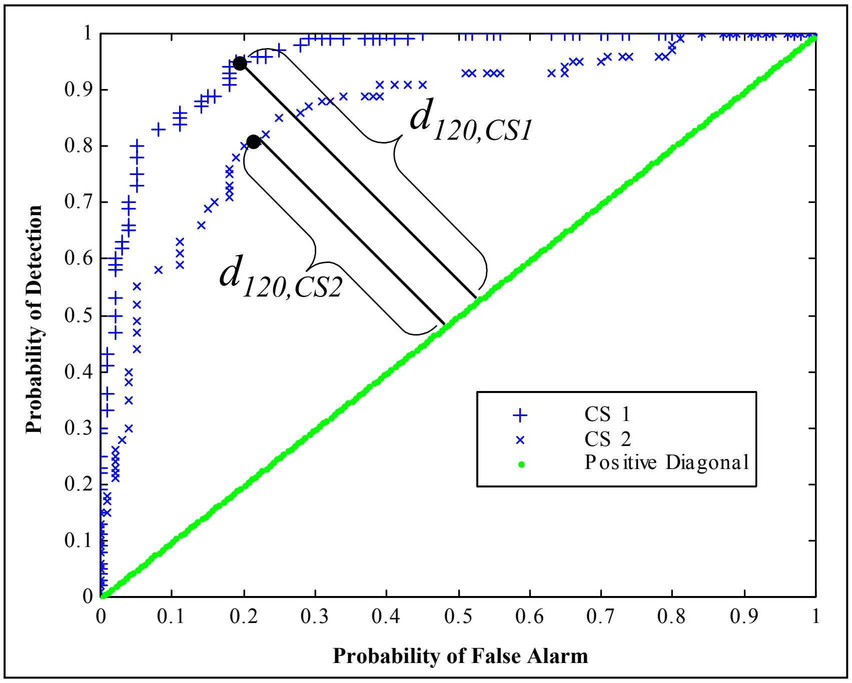


Figure 2.21 Depiction of Average Metric Distance Metric Calculation.

A similar metric introduced in biostatistical research is that of the q-norm metric, which compares the distance between matched ROC curves (13). The second method measures the distances based upon a line with a slope $b = -\sqrt{(m/n)}$ where *m* and *n* are the number of samples taken from the target and non-target distributions. Though this metric does not require that the same number of non-targets and targets be sampled, the use of differently sized target and non-target populations complicates this metric.

Consider a simple example of the average metric distance metric for comparing two separate ATR CSs. The ROC curves for the CSs have been provided in Figure 2.22. Notice that CS_2 seems to be the superior system, and, in fact, it is by design of the target distributions. CS_1 is based upon a *N(3,2)* target distribution, while CS_2 is based upon a *N(3,1)* target distribution. However, the ROC curves cross, which, by recommendations from the biostatistical community, indicates that the use of the *AUC* measure is unreliable. Using the Euclidean distance metric, an average metric distance measure may be calculated to compare each CS to the negative diagonal line. The results are presented in Table 2.8. For this example, the results indicate that CS_2 is the better classifier, which we know to be true.

Table 2.8 Comparison of *AUC* and *Average Metric Distance* Measures.

Classifier	*AUC*	*Avg Metric Dist (q=1)*	*Avg Metric Dist (q=2)*	*Avg Metric Dist (q=100)*
1	0.8850	0.2071	0.1464	0.1043
2	0.9231	0.1797	0.1271	0.0905

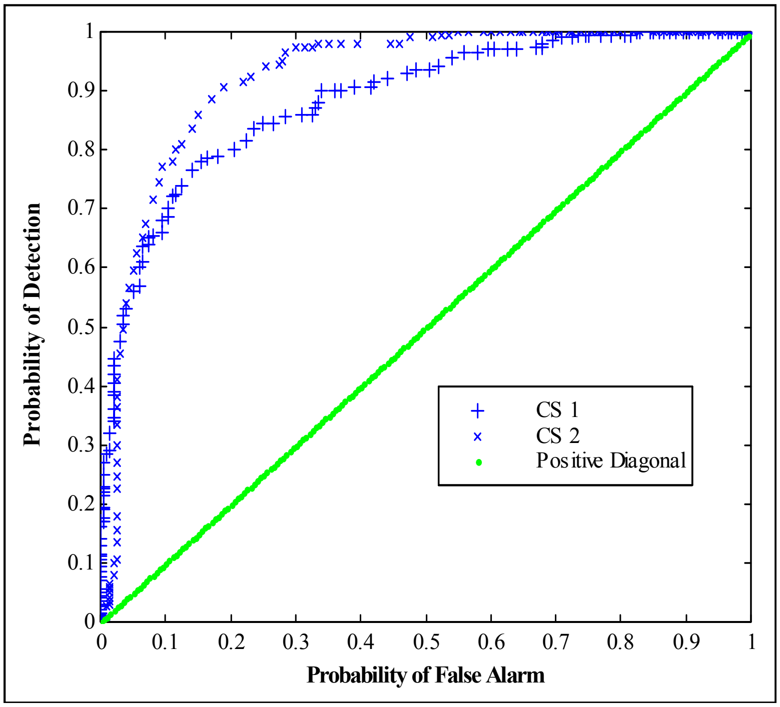


Figure 2.22 ROC Curves for Average Metric Distance Example.

A final method of comparing ROC curves lies in the construction of confidence bands around the ROC curve. Non-parametric confidence bands based on the Kolmogorov theory concerning distributions can be constructed about each point on an empirical ROC curve (13). Thus, the $(1-\alpha)$ Kolmogorov-Smirnov (K-S) confidence band takes the form $(F_m(t) - d, F_m(t) + d)$ on a ROC curve, where $F_m(t)$ represents a realized target probability of detection value at threshold t and d is the half-length distance measure of the given confidence band. Using $G_m(t)$ and e to represent the non-target probability of false alarm value and confidence band half-length, the overall confidence rectangles about a single point on the ROC curve takes the form:

$$P\{ F_m(t) - d < F_m(t) < F_m(t) + d,\ G_m(t) - e < G_m(t) < G_m(t) + e\} = (1-\alpha)^2\ , \quad (2.44)$$

assuming the independence of the two distributions (13). Thus, the collection of rectangles with width $2d$ and height $2e$ each centered at an observed point of the ROC curve has simultaneous coverage $(1-\alpha)^2$. Note, this coverage is valid at all thresholds since the sensitivity and specificity are the same on the interval $[t_i, t_{i+1})$ as at t_i. The confidence statement about this collection of confidence rectangles is merely that, for all thresholds simultaneously, the theoretical values of (1-specificity) and sensitivity are in their associated rectangles with confidence $(1-\alpha)^2$. Hypothetically, these bands could be used in similar fashion to confidence intervals about point estimates for the purposes of CS comparison. The procedure for constructing these bands is as follows:

1. Generate the empirical ROC curve by plotting the probability of false alarms and probability of detection from the target and non-target distributions.

2. Construct the confidence rectangle about each point in the empirical ROC curve. Using the critical values for the K-S Goodness-of-Fit Test for a single sample and a desired confidence level, α. For observations greater than 40, the formula $K/\sqrt{n}$ is used to compute the confidence bands. Let m refer to the observations of the target distribution and n represent observations of the non-target distribution.

As an example of this procedure, a randomly selected set of targets (m = 200) generated from a Normal distribution ($\mu = 4$, $\sigma^2 = 1.5$) and non-targets (n = 200) generated from a Normal distribution ($\mu = 0$, $\sigma^2 = 2$), seen in Figure 2.23, are used to create an empirical ROC curve, shown in Figure 2.24.

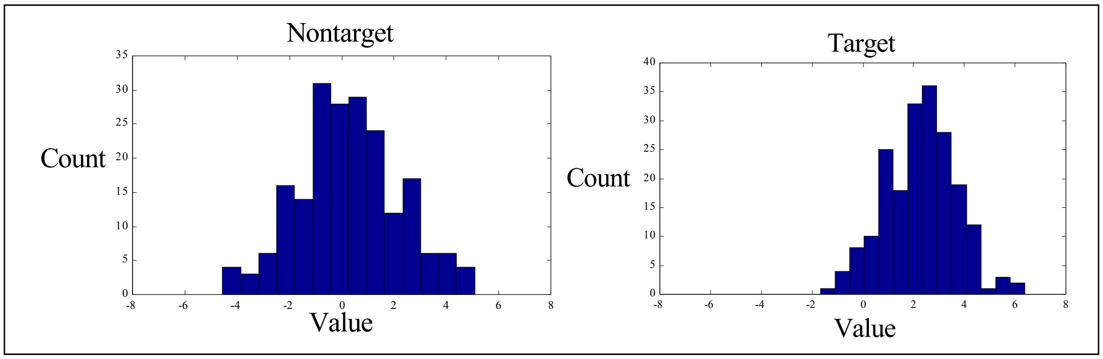


Figure 2.23 Empirical Data Set: Target~N(4,1.5) and Non-target~N(0,2).

A confidence rectangle is then constructed about each point in the ROC curve. In this instance, the rectangle is a square due to the equal number of observations from each distribution, e.g. $n = m = 200$. Thus, for $\alpha = 0.05$, the half-length of each confidence rectangle is equal to $d = e = 1.36/\sqrt{m}$. The resultant confidence bands can be seen in Figure 2.24.

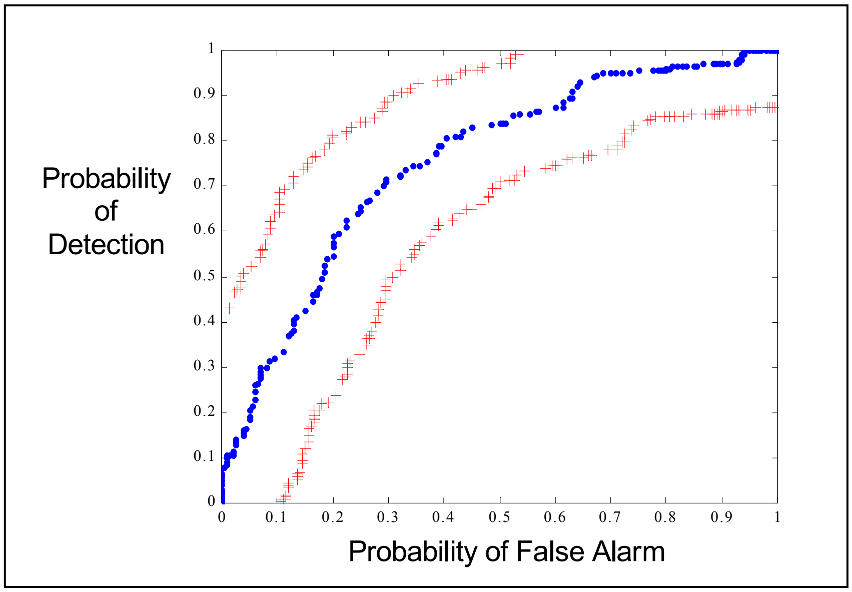


Figure 2.24 ROC Curve and Associated K-S Confidence Rectangles (*m*=*n*=200).

Figure 2.25 illustrates the change in size of the K-S confidence rectangles when the number of observations of the non-target population increases to 1000. Note that the confidence rectangles have adjusted in width.

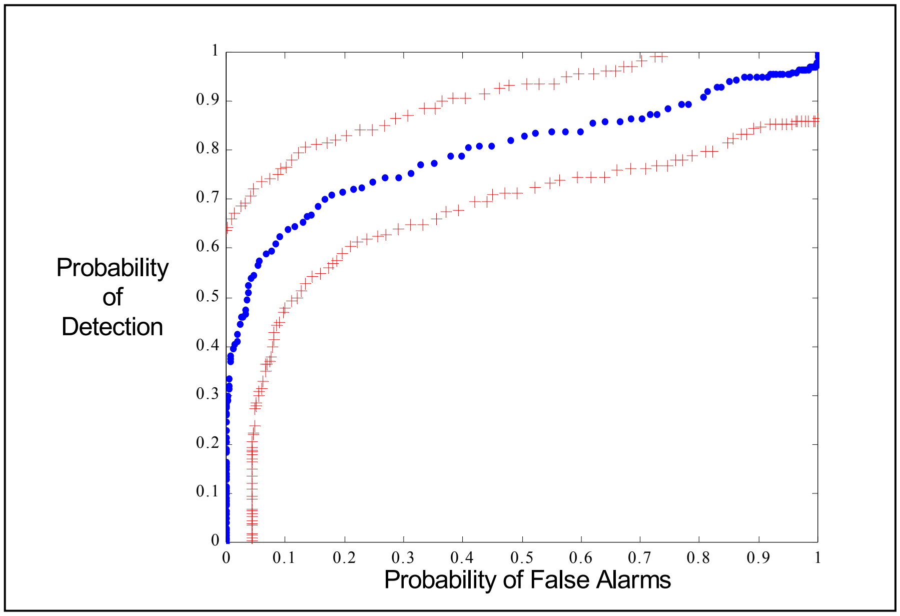


Figure 2.25 ROC Curve and Associated K-S Confidence Rectangles (*n* = 1000).

2.3.3.4 Multinomial Selection Procedure (MSP).

Multinomial selection procedures (MSPs) have only recently been applied to the evaluation of competing algorithms (4,5). Multinomial selection problems involve the comparison of *k* classification systems across a given objective performance measure (8,9). MSP may also compare *k* systems across *n* classes, rather than being limited to the target/non-target alternatives in the typical DUSD-BM scenario (8,9). The objective of the MSP is to find the system, given a limited amount of data, which is most likely to be the best performer in a single trial among the systems, rather than identifying the best average performer over the long run (8,9). Thus, instead of relying on the overall performance power of *AUC*, the MSP generates a metric on a CS's performance per class with which to compare systems.

Consider an example of the MSP in the performance comparison between three notional CSs classifying a two-class data set of the classic XOR problem. Each CS has been trained using the same data set, balanced between the two classes. In this example, there are 250 data points (125 for Class 1 and 125 for Class 2) for training each classifier, and the same number of points for testing each classifier. A plot of the data to be classified is given in Figure 2.26. For this example, three classifiers are examined: a linear statistical classifier, a quadratic statistical classifier, and a multi-layer perceptron (MLP) artificial neural network classifier. The idea is to use the classification results of the three different classifiers. The classification accuracy and confusion matrix for each classifier is listed in Table 2.9. Notice that though the linear classifier proves to be inferior, there is no clearly superior classification system. This inability to distinguish between the classifiers leads to the use of the MSP for a better performance metric.

Procedure BEM (Bechhofer, Elmaghraby, and Morse) is a classical solution procedure for the MSP (12). On the assumption that larger is better, BEM selects the system having the largest value of the performance measure in more replications than any other, as the best system. Another necessary assumption is that for a multinomial distribution there is a constant probability of success over all test trials. This assumption holds as long as the test trials are at random, and the probabilities of success obtained are still estimates of the probabilities of winning in any randomly selected trial (7). A modified version of the BEM procedure is given below:

1. Given v_j Class j test data points, compare estimated posterior Class j probabilities for each classifier.
2. Select the best classifier for each data point as the classifier with the maximum estimated posterior Class j probability.

3. Compute the number of wins/successes $Y_{i|j}$ for each classifier *i* given Class *j* data.

4. Let $Y_{[1|j]} \leq Y_{[2|j]}$ be the ranked number of successes from Step 3. Select the classifier associated with the largest count, $Y_{[3|j]}$, as the best for Class *j*.

Using this technique, a point estimate can be computed for the conditional probability $P(C_i|\Phi_j)$ of each classifier C_i being the best given the class Φ_j using

$$P(C_i \mid \Phi_j) = \frac{Y_{i|j}}{v_j} = \hat{p}_s = P_{BEST\ BY\ CLASS}. \tag{2.45}$$

This is accomplished when the number of successes *Yi|j* for each classifier *I* given *vj* Class *j* test data points, is modeled as a single multinomial distribution. The corresponding confidence intervals can be constructed using:

$$\hat{p}_s \pm Z_{1-\frac{\alpha}{2}} \sqrt{\frac{\hat{p}_s(1-\hat{p}_s)}{n}}\ . \tag{2.46}$$

The total probability that each classifier is the best according to the estimated posterior probabilities can be computed using the law of total probability:

$$P(C_i) = P(C_i \mid \Phi_1)P(\Phi_1) + P(C_i \mid \Phi_2)P(\Phi_2) = P_{BEST\ OVERALL} \tag{2.47}$$

where P(Φj) are the prior probabilities for each class (in the example, these prior probabilities are equal to 0.5). Tables 2.10 and 2.11 illustrate the use of the BEM procedure with the given classifiers for both Class 1 and Class 2 data, respectively. Table 2.12 provides the resultant confidence intervals ($\alpha = 0.05$) around the P_{BEST} point estimates of the BEM procedure for each classifier.

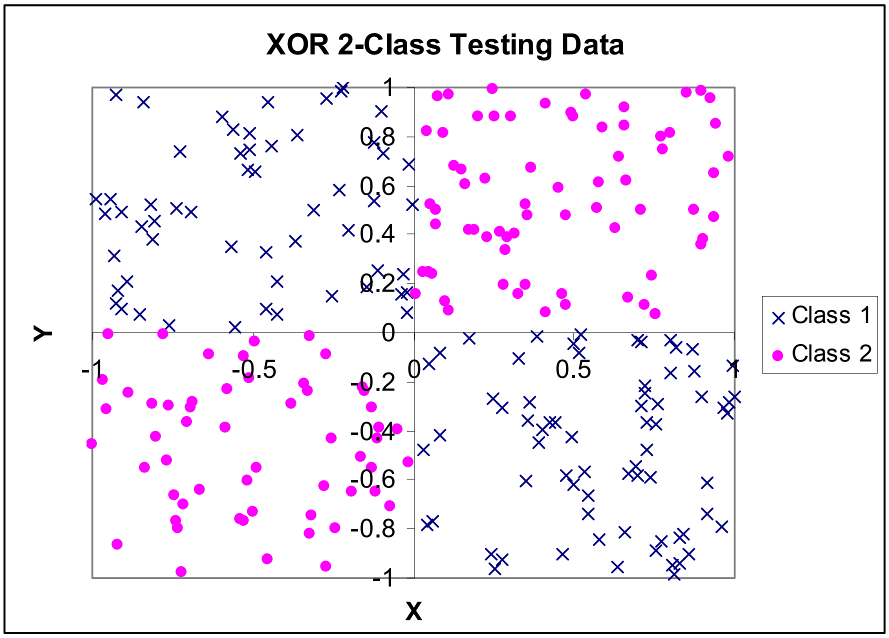


Figure 2.26 Testing Set of Two-Class XOR Data (250 Samples).

Table 2.9 Confusion Matrices and Classification Accuracies for MSP Example.

		Linear Classifier Classified As			Quadratic Classifier Classified As			MLP Classifier Classified As	
		C1	C2		C1	C2		C1	C2
Actual	C1	54	71	C1	113	12	C1	113	12
(Truth)	C2	64	61	C2	9	116	C2	11	114
CA		0.460			0.916			0.908	

Table 2.10 BEM Procedure Illustrated for Class 1 XOR Data.

Test Data	Posterior Probabilities			Win/Successes = 1		
Number	Linear	Quadratic	MLP	Linear	Quadratic	MLP
1	0.7473	0.7229	0.3510	1	0	0
2	0.3025	0.5257	0.8896	0	0	1
3	0.6384	0.5819	0.1328	1	0	0
…	…	…	…	…	…	…
125	0.3652	0.4338	0.8673	0	1	0
				Linear	Quadratic	MLP
	Successes ($Y_{j\|1}$)=			4	18	101
				3.2%	46.4%	50.4%

Table 2.11 BEM Procedure Illustrated for Class 2 XOR Data.

Test Data	Posterior Probabilities			Win/Successes = 1		
Number	Linear	Quadratic	MLP	Linear	Quadratic	MLP
1	0.4512	0.8705	1.000	0	0	1
2	0.6974	0.8442	1.000	0	0	1
3	0.6724	0.9265	1.000	0	0	1
…	…	…	…	…	…	…
125	0.4489	0.9942	0.9489	1	0	0
				Linear	Quadratic	MLP
	Successes ($Y_{j\|1}$)=			3	34	88
				2.4%	27.2%	70.4%

Table 2.12 Procedure BEM P_{BEST} Estimates and 95% Confidence Intervals for XOR Data.

P_{BEST}	**Linear**	**Quadratic**	**MLP**
Class 1	0.032	0.464	0.504
CI	[0.001 0.063]	[0.377 0.551]	[0.416 0.592]
Class 2	0.024	0.272	0.704
CI	[0 0.051]	[0.194 0.350]	[0.624 0.784]
Both Classes	0.028	0.368	0.604
CI	[0.000 0.029]	[0.283 0.453]	[0.518 0.690]

The results show that for classifying Class 1 data, the MPL and quadratic statistical classifier are similar, but when classifying Class 2 data, the MLP classifier is statistically the best classifier by using the MSP P_{BEST} metric. Finally, the confidence intervals indicate that the MLP classifier is the best overall system for classifying the given XOR data set. Results from various pattern recognition problems have indicated that the MSP can be used to distinguish differences between CSs that other performance measures, such as *AUC* and *CA*, cannot (4,5).

2.3.3.5 Linear Goal Programming (LGP).

Linear goal programming (LGP) is a constrained optimization technique used by decision-makers to solve multivariable, multigoal problems (2). Many problems, such as comparison of ATR CSs, involve not only multiple objectives, but also multiple *conflicting* objectives (2). All LGP models consist of three components: an objective function, goal constraints, and non-negativity requirements (2). Differences from the typical linear programming model include the preemptive priority factors, deviational variables, and the concept of *satisficing*. Preemptive priority factors (P_k) is a method whereby goals are ranked ordered, where P_1 represents the most important goal and P_K represents the least important goal of K goals. Thus, the P_k are numeric values that represent the decision-maker's goal priorities within the model. Deviational variables, denoted d_i^- and d_i^+, express the deviation from a particular goal as the LGP is solved. Finally, the idea of *satisficing* implies that the LGP will seek a solution that satisfies as many goals as possible rather than optimizing a single goal (2). Thus, proper goal selection is an important aspect to LGP formulation.

In LGP, there are three different types of objective functions and six different types of goal constraints. Table 2.13 lists the different types of objective functions and their purposes. Table 2.14 lists the different goal constraints and their uses. LGP formulation involves the following six steps (2):

1. ***Define the decision variables.*** Clearly state what the unknown decision variables are. In an ATR CS comparison application, this may be the various CSs that are being subject to comparison.

2. ***State the goal constraints.*** Identify the right-hand-side variables first paying attention to the permissible deviation for each deviational variable.

3. ***Determine the preemptive priorities***. Rank the goals in accordance with the decision-maker's stated preference. For CS comparison, this could be the mission scenario or the desired performance measure to be maximized or minimized.

4. ***Determine the differential weights.*** Examine preferences within a specific goal level. For instance, a goal range may be given where any deviation outside of that range affects the decision.

5. ***State the objective function.*** Select the correct deviational variable for inclusion in the objective function. Ensure the that the deviational variables correspond to the appropriate preemptive priority factor.

6. ***State the non-negativity requirements.***

Table 2.13 LGP Objective Functions.

Objective Function	Purpose
Minimize $Z = \Sigma\ (d_i^+ + d_i^-)$ for all I	Used when deviational variables are not distinguished by priority or weighting

Minimize $Z = \Sigma\ P_k(d_i^+ + d_i^-)$
for all i
for $k = 1,2,\ldots K$
	Used when K goals are ranked by P_k priorities; Goals are ranked but deviational variables are of equal importance

Minimize $Z = \Sigma\ P_k w_{kl}(d_I^+ + d_i^-)$
for all i
for $k = 1,2,\ldots K$
for $l = 1,2,\ldots,L$
	Used when K goals are ranked by P_k priorities; Goals are ranked and deviational variables are weighted by the w_{kl} differential weighting

Table 2.14 LGP Goal Constraint Types.

Goal Constraint	Deviational Variable in Objective Function	Possible Deviation	Unrestricted Deviations	Desired Usage of Right Hand Side Value
$a_{ij} + d_i^- = b_i$	d_I^-	Negative	None	Equal to b_i
$a_{ij} - d_i^+ = b_i$	d_I^+	Positive	None	Equal to b_i
$a_{ij} + d_i^- - d_i^+ = b_i$	d_i^-	Negative and Positive	Positive	b_i or more
$a_{ij} + d_i^- - d_i^+ = b_i$	d_i^+	Negative and Positive	Negative	b_i or less
$a_{ij} + d_i^- - d_i^+ = b_i$	d_i^- *and* d_i^+	Negative and Positive	None	Equal to b_i
$a_{ij} - d_i^+ = b_i$	d_i^+ *(artificial)*	None	None	Exactly b_i

The following example illustrates how a LGP model can be used to compare between two ATR CSs. Suppose the goal of comparison is to determine which CS is more suitable for a CID mission. A CID mission requires a low probability of false alarm since the goal is to destroy known enemy targets. The most important priority in the CID mission is to minimize the number of possibly civilian non-targets. Thus, the value of p_{FA} in the objective function is appropriately weighted. In fact, the decision maker has decided that this performance measure is 3 times more important than any other measure. A way to incorporate the other dimensions of performance for a CID mission is to include the deviations from p_{ID}, p_{CC}, or p_D in the objective function. Minimizing p_{FA} without doing so allows the LGP to select an ATR CS that may not detect anything. The factors of CS cost, computational time, and other various measures, though not used in this example, could be incorporated into an expanded model.

Suppose that an ATR CS, denoted C_1, has a $p_D = 0.560$, a $p_{FA} = 0.440$, a $p_{CC} = 0.762$, and a $p_{ID} = 0.833$. Further suppose that another ATR CS, C_2, has a $p_D = 0.680$, a $p_{FA} =$

0.540, a p_{CC} = 0.804, and a p_{ID} = 0.922. Assuming that is the only available information concerning the two systems, the LGP formulation could be given as follows:

$$\text{Minimize} \qquad Z = 3(d_1^{+}) + (d_2^{-} + d_3^{-} + d_4^{-}) \tag{2.48}$$

$$\text{subject to:} \quad 0.440x_1 + 0.540x_2 - d_1^{+} = 0 \qquad (p_{FA} \text{ goal constraint}) \tag{2.49}$$

$$0.560x_1 + 0.680x_2 + d_2^{-} = 1 \qquad (p_D \text{ goal constraint}) \tag{2.50}$$

$$0.762x_1 + 0.804x_2 + d_3^{-} = 1 \qquad (p_{CC} \text{ goal constraint}) \tag{2.51}$$

$$0.833x_1 + 0.922x_2 + d_4^{-} = 1 \qquad (p_{ID} \text{ goal constraint}) \tag{2.52}$$

$$x_1 + x_2 = 1 \qquad \text{(constraint which forces CS selection)} \tag{2.53}$$

$$x_1, x_2 \in \{0,1\} \quad \text{(assignment constraint)} \tag{2.54}$$

$$d_1^{-}, d_2^{-}, d_3^{-}, d_4^{-} \geq 0 \;\; \text{(non-negativity constraint)} \tag{2.55}$$

Notice that P_1 equals 3 and P_2 equals 1 since the p_{FA} goal is three times as important as the other goals. The three goal constraints associated with preemptive priority factor P_2 share the same preemptive priority factor due to the fact that they have no distinguishable priority over each other. These values are gathered from discussions with the decision maker and may be arrived at through a value hierarchial analysis. Also notice that the decision variables, x_1 and x_2, correspond to the two separate ATR CSs, C_1 and C_2.

The deviational variables, d_i^{+} and d_i^{-}, represent the amount over or under the ith goal constraint when a particular ATR CS is chosen. For instance, in the p_{FA} goal constraint, the object is to have the lowest possible p_{FA} while maintaining all other goals, if possible. Thus, the right-hand-side (RHS) of the p_{FA} goal constraint is set to zero. The lowest possible deviation (d_1^{+}) above 0 is the ideal solution for that particular goal, which is evident in the minimization aspect of the objective function. The opposite is true of the remaining deviational variables since the problem formulation attempts to minimize the amount under 1, the RHS of the remaining goal constraints, which an ATR CS forces upon the solution. As a final aspect to the problem, the decision-maker made no mention of an acceptable p_{FA}. Though an acceptable CS was chosen by the LGP solution, the CS itself may fall short of an overall mission requirement not mentioned in the problem description. However, since such constraints are easily added to an LGP formulation, the burden of correctly formulating all requirements and constraints for a given LGP solution falls on the analyst.

The solution to this simplified problem is the selection of C_1 as the appropriate CS for use in the given mission. However, the objective function value is 2.17 for the selection of C_1 and 2.21 for the selection of C_2. This very slight difference has no statistical significance for choosing one CS over the other. Thus, C_1 has a very slight advantage with its low p_{FA}, but C_2 almost overcomes its deficiencies in spite of the decision-maker's preference for a low probability of false alarm.

For the typical ATR CS comparison scenario, the decision variables, which correspond to N CSs to be compared, are binary. Thus, selection of C_1 in the previous example results in x_1 equal to 1 while not selecting C_2 results in x_2 equal to 0. In other words, there is no way to choose x_1 equal to 0.5 and x_2 equal to 0.5 since only one CS can be used by an aerial platform during its mission. One technique in LGP to use in this case is the *branch and bound* method for integer solutions (2).

One hindrance in using the single LGP formulation for the given scenario is that the performance measures of the ATR CSs are point estimates of the actual values. Thus, confidence intervals may be constructed around the point estimates in order to estimate the variance in the point estimates. For the scenario in question, assume that there are 75 known targets and 50 known confusers. Thus, C_1 correctly detected 42 of the 75 targets and incorrectly declared 22 of the 50 confusers. C_2, on the other hand, correctly detected 51 of the 75 targets but incorrectly declared 27 of the 50 confusers. From these results, it should be clear that C_1 correctly classified 32 and correctly identified 35 of the 42 detected targets. Similarly, C_2 correctly classified 41 and correctly identified 47 of the 51 detected targets. Using confidence intervals like those detailed in section 2.2.1.4, the 95% confidence intervals are generated and presented in Table 2.15.

Table 2.15 Point Estimates and 95% Confidence Intervals for LGP Example.

	C_1			C_2		
Measure	**Lower Level**	**Point Estimate**	**Upper Level**	**Lower Level**	**Point Estimate**	**Upper Level**
P_{FA}	0.302	0.440	0.578	0.402	0.540	0.678
P_D	0.448	0.560	0.672	0.574	0.680	0.786
P_{CC}	0.633	0.762	0.891	0.699	0.804	0.913
P_{ID}	0.721	0.833	0.946	0.848	0.922	0.995

Having quantified the possible variance within each point estimate, it is evident that a single iteration of the LGP formulation will not suffice. There are at least two approaches to assessing the variability within the given LGP formulation. The first is to solve an LGP of each permutation of performance measures at the lower level, upper level, and mean values. This results in 3^8 (6561) different LGP solutions. Table 2.16 lists the results from this approach. From the results, it is clear that C_1 is the superior system.

Table 2.16 LGP Results at Tri-Level Performance Measure Values (6561 Reps).

Classifier	Wins	Percentage	Mean Z
C_1	3500	0.5335	2.1647
C_2	3061	0.4665	2.2130

The second approach of comparing the two systems is to generate a Monte Carlo simulation of the point estimates of the probabilistic performance measures. By using a random number generator, the point estimates are randomized and contain more variance than the previous method where the lower and upper bounds represented the greatest deviation from the point estimate. The randomized estimates are then fed into the LGP formulation to produce an objective function value. One benefit of this technique is that the number of replications performed is limited only by computational time and the random number generator used. For the example, 10,000 replications were generated and each performance measure point estimate was distributed as a Normal distribution with a mean and variance based upon the data of Table 2.15. The results, seen in Table 2.17, are very similar to the previous method. The results indicate that C_1 is a superior system, but with a smaller margin of difference.

Table 2.17 Monte Carlo LGP Formulation Results (10,000 Reps).

Classifier	Wins	Percentage	Mean Z
C_1	5570	0.557	2.1664
C_2	4430	0.443	2.2151

The major advantages of the goal programming method are that all performance measures of an ATR CS could conceivably be incorporated into the LGP formulation of

CS comparison and that the decision-maker has control over the goal priority selection. Thus, each LGP formulation is tailored to the decision-maker's objectives and uses the greatest amount of information possible to influence a decision. Another benefit to LGP is the ability to quantify a system's inability to meet a given desired performance level. The LGP can be formulated to reflect whether or not a given ATR CS can even meet the desired performance levels.

The disadvantages of the multiple replication techniques hinge upon the large amount of computational time, but more importantly, the reliance upon the initial point estimates used in the LGP formulation. The point estimates used when building these comparisons should be well defined. In other words, the performance measure point estimates must be based upon a reasonable number of replications that allow an analyst to make assumptions concerning the distribution of the performance measure. Without this, the utility of the Monte Carlo and parameter levels (mean, upper bound, lower bound) comparison techniques are somewhat limited despite the large number of replications in each.

Overall, the LGP formulation depends on the preferences of the decision-maker. Each parameter used within the LGP must make sense to the decision-maker for use in addressing the ultimate goals of the decision to be made. Thus, if the LGP method does not mirror the objectives listed in the decision-maker's fundamental objective hierarchy, then the results will be rather useless. For instance, if a decision-maker's objective is to find the classifier which best detects the targets in a SAR image, then the probability of detection should most likely be the performance measure which is weighted most heavily in the calculations.

2.3.3.6 Decision Analysis (DA).

Since decisions are at the heart of ATR evaluation, a review of decision analysis (DA) methods and techniques is fitting. The deficiencies in current techniques for comparing performance measures in ATR CSs are well known (7). For instance, the direct comparison of probability performance measures is generally inadequate as these metrics provide information for only one decision threshold. The *AUC* ROC performance measure solves this problem, but has been shown to be unreliable in the comparison of very similar systems (7,17,36,43,46). In the biomedical analysis community, it is considered bad practice to compare the *AUC* for ROC curves that overlap (17,43). The *AUC* for two CSs may be significantly different, but the implementation of the CSs may provide similar risk/benefit results and vice versa (17,46). A DA approach could allow a decision-maker to quantify the risks and benefits of using two separate ATR CSs, compare the expected performance, and make the appropriate decision.

The field of decision analysis focuses on decisions where preferences and uncertainties need to be modeled. The concept is to present a decision maker with a list of alternatives and their expected impact from which he or she may make a more informed decision (19). The decision maker provides input as to the risks that he or she is willing to accept and anxious to avoid by indicating outcome preferences. The benefits and drawbacks of each alternative are modeled in order to describe the problem at hand. Finally, the decision maker along with experts associated with the problem may provide insight into the uncertainties involved with making a decision, which are also modeled within the problem formulation. Thus, subjective judgments may be included in the formulation of a DA approach to decision making. While the incorporation of personal judgments is necessary, these judgments should not be considered perfect. Personal insights on uncertainty and system performance value can be misleading or limited (19). Therefore, DA techniques must be applied with care.

The concept behind DA is to use the known information about the problem at hand, provide expert opinion or standards to the uncertainty in the problem, and quantify the value of making a particular decision. The DA process as a whole begins with a description of the problem. Once the appropriate problem has been identified, the objectives and alternatives within the problem must be identified. The next step is to decompose the problem into the structure of the decision, measure the unknown elements of the problem, and obtain the preferences or restrictions of the decision maker (19). When these elements of model formulation are complete, the best alternative should be selected. Sensitivity analysis may then be performed on the decision to determine if small changes in the aspects of the model result in large changes in the outcome of the decision or even change the optimal decision. If so, the decision maker may wish to reevaluate his or her decision. If analysis indicates that the decision should be changed or more alternatives should be examined, then the modeling process begins again (19). Figure 2.27 depicts the DA cycle.

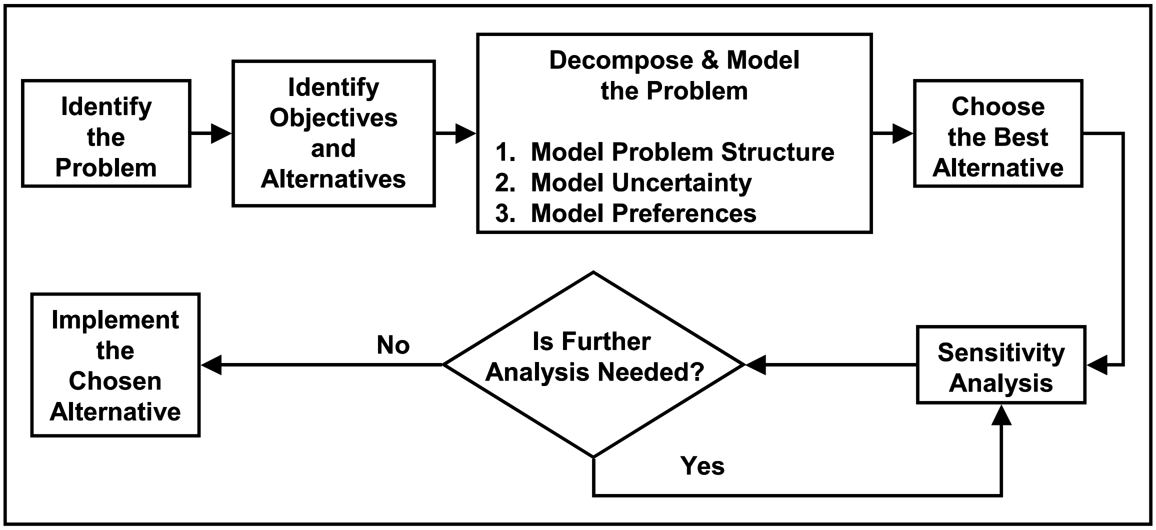

Identify the Problem
Identify Objectives and Alternatives
Decompose & Model the Problem
1. Model Problem Structure
2. Model Uncertainty
3. Model Preferences
Choose the Best Alternative
Sensitivity Analysis
Is Further Analysis Needed?
No
Yes
Implement the Chosen Alternative

Figure 2.27 Decision Analysis Process Flow Chart (19).

Modeling the problem is an important step that may immediately help a decision maker. Influence diagrams provide a graphical approach to capturing each element that has an impact upon the decisions to be made or is impacted by the decisions after they are made. Shapes within the influence diagram, or nodes, model the decisions, chance events, and value of outcomes (19). In the influence diagram context, decisions are represented by squares, ovals represent problem uncertainty, and rectangles with rounded corners depict values. Arrows, or arcs, indicate the relationship between the given nodes. As an example, suppose that a decision maker with an ample amount of capital must decide whether or not to invest in an emerging technology. The decision depends upon the future success of the given technology. If the decision maker invests and the venture succeeds, the result is a large monetary return. If the venture fails, then the investor loses all of his capital. The decision maker could also choose not to invest and merely keep the capital. Figure 2.28 illustrates a simple influence diagram for the decision of whether or not to invest in a new technology.

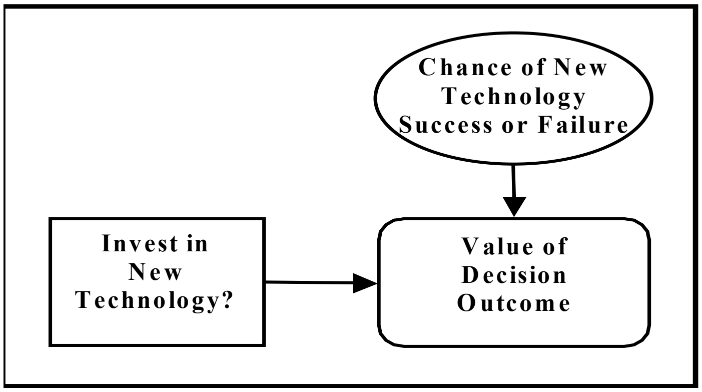

Chance of New Technology Success or Failure
Invest in New Technology?
Value of Decision Outcome

Figure 2.28 Influence Diagram of New Technology Decision Problem (19).

Decision trees are a popular method of modeling the problem by portraying the structure of the decisions, events, and uncertainties. Decision nodes, represented by squares, indicate where decisions are to be made within the problem. Uncertainty nodes, depicted by circles, represent the unknown factors that impact the problem, such as events that may happen following a decision (19). The events branching from an uncertainty node must be mutually exclusive (only one branch may be taken) and collectively exhaustive (no other alternatives for that node exist) (19). At the end of the branch of a decision tree is the cost or value associated with the decision or outcome. The decision tree, typically viewed as a time progressing model from left to right, represents all possible paths that a decision maker may experience through the decision making process of a problem. The time-oriented structure of a decision tree is important in that uncertainty nodes to the left of a decision node indicate that the decision depends on an unknown outcome within the problem domain. A decision node to the left of an uncertainty node indicates that a decision is made and the uncertain events follow (19). Continuing from the previous technology investment example, further suppose that for an investment of $2M the possible investment return would be $10M if the venture succeeds. Experts expect that the technology has a 40% chance of being successful. This information can be entered into the form of a decision tree. Figure 2.29 depicts a simplistic decision tree for the decision maker deciding whether or not to invest in a new technology.

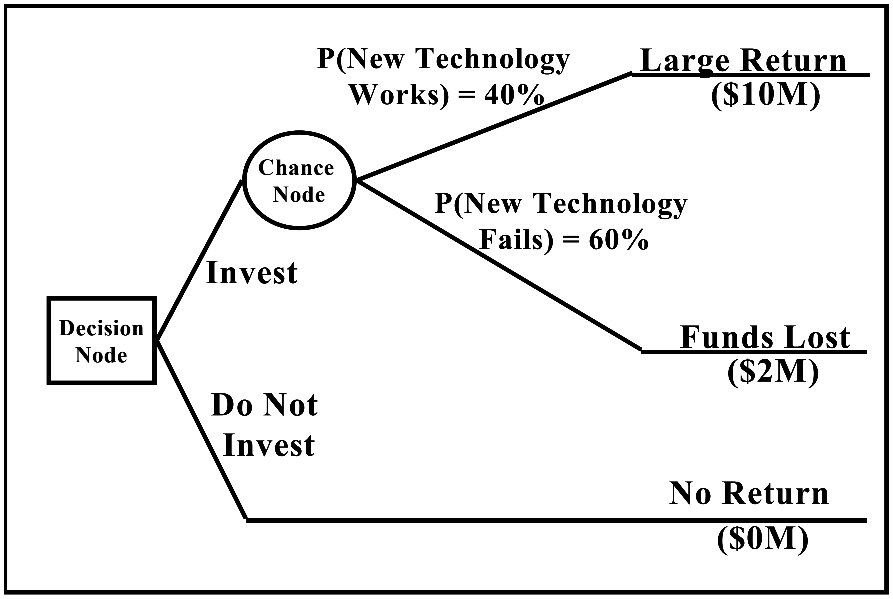


Figure 2.29 Decision Tree Representation of New Technology Decision Problem.

Decision trees and influence diagrams are the primary modeling tools used in the DA problem solving approach. Though not much information is displayed in influence diagrams, their strength is their simplistic representation of the problem. Introducing a decision maker to the DA approach through influence diagrams is usually the preferred method. Decision trees, on the other hand, depict each possible outcome and decision associated with the problem. While this approach is helpful to the analyst and excellent for displaying detailed information, decision trees tend to get "messy" for large, complex problems. However, the decision should not be to use one over the other since they are isomorphic. In other words, a decision tree can be constructed from a well- built influence diagram and vice versa (19).

Modeling uncertainty can be a very difficult task. Quantifying a person's feelings or beliefs about an uncertain aspect of a problem requires a degree of precision to make the solution useful. The primary method of modeling chance events in a DA model is through the use of probability (19). Since events preceding or following a decision may or may not happen with a certain probability, it is possible to model these events through probability models. These probability models may be based upon known probability density functions, such as the normal or exponential functions, subjective probability estimation, historical data, or through data generated from Monte Carlo simulations.

Modeling the decision maker's preferences is important in that almost every decision involves a trade-off. It is possible to model a decision maker's risk policy by developing a decision utility function. While a decision may be made based upon the optimal expected outcome, a different alternative on the same decision may be made based on other factors associated with the decision (19). For instance, consider the game proposed in Figure 2.30 where a player has the option to play one of two games (G_1 and G_2) that offer differing expected values over the long run. In both games, the player flips a fair coin. In G_1, if the result is "heads" the player wins \$30, while the player must pay \$1 if "tails" is the outcome. The expected long run value of G_1 is \$14.50. G_2, however, pays the player \$2000 for "heads", but requires a payment of \$1900 for a "tails" result. The expected value for G_2 is \$50, which is superior to G_1 over the long run. Under ordinary decision rules, the player would choose G_2 to player in order to gain more winnings in the long run. However, the decision dynamics change if each game could only be played once. In this case, the range of possible outcomes drives the decision. The best case is for the player to choose G_2 and win \$2000. However, in playing G_2, the player risks losing \$1900. Most people would choose playing G_1 and risking only \$1 with the possibility of winning \$30 instead of risking such a large amount of money in a game of chance (19). Thus, this situation implies that a decision maker must incorporate a risk policy into the formulation of the problem model. This can be accomplished through the use of preference modeling.

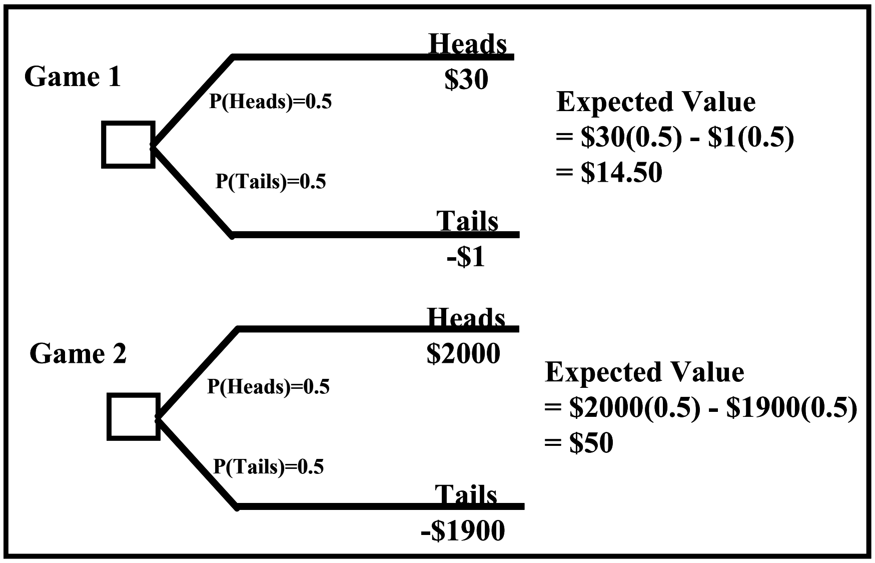


Figure 2.30 Risk Gaming Example.

Utility functions capture the relationship between the value of an outcome and the risk the decision maker is willing to accept (19). Different people respond to risk differently. Therefore, each decision maker has a personalized risk attitude utility function. A risk-seeker, or gambler, is one who is inclined to accept risk in return for a return. This type of person may elect to play one round of G_2. The chance of winning \$2000 outweighs the risk of losing \$1900 in a risk-seeker's mind. A risk-averse individual attempts to avoid risk. This type of person may stick with G_1 in the long run to avoid the loss of \$1900 in a single coin flip. A risk-neutral person ignores the effect of risk in a decision. For this individual, maximizing the expected value of a decision is the same as maximizing the expected utility of the same decision (19). Figure 2.31 illustrates the shape of a utility function for the risk-seeking, the risk-averse, and the risk-neutral.

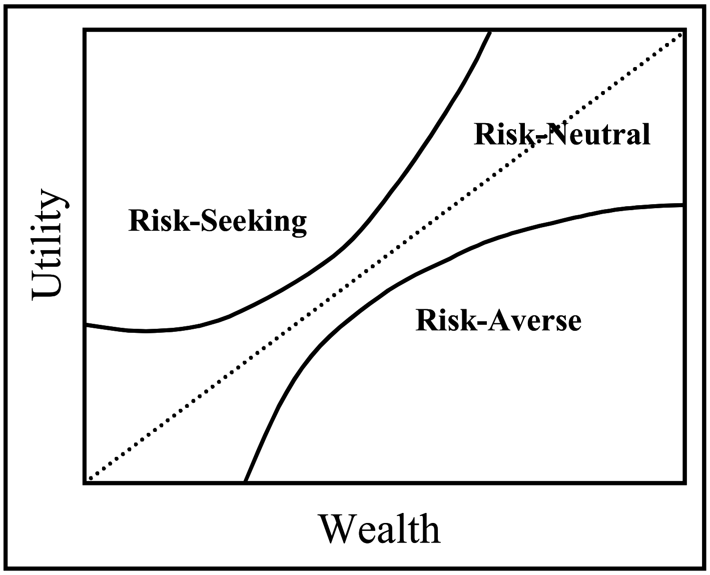


Figure 2.31 Three Different Shapes for Utility Functions (19).

Advantages of the DA approach to comparing ATR CSs include the ability to compare multiple performance measures, the inclusion of decision maker standards and expert opinion, and the ability to perform sensitivity analysis on decision outcomes. Despite these benefits, the DA approach does have its limitations. The input from decision makers or field experts can be biased, misleading, or difficult to obtain. Also, placing a value on the outcome of a decision can be very difficult, especially if human life is involved. Finally, the use of DA techniques does not guarantee a good decision. The purpose of DA is to provide an *informed* decision that is based on the *expected* return (19).

III. A Decision Analysis-Based Automatic Target Recognition Evaluation Methodology

3.1 Overview

The goal of this research is to create a credible, defensible method for making decisions concerning ATR technology development while using all of the pertinent measures of performance (MOPs) and measures of effectiveness (MOEs), including all of the interested parties, and incorporating the preferences and values of the decision-makers. In regards to the interested parties, the role of the ATR evaluator is to oversee ATR evaluation at the research and development level, but the evaluator may or may not be an expert on what performance levels a fielded ATR system may require. The warfighter, on the other hand, will ultimately use the ATR end product in an operational environment and should understand the impact of ATR operational effectiveness, but may or may not fully understand the intricacies of ATR performance assessment, particularly during the testing phase. Thus, the concept is to construct two separate decision analysis frameworks: one for the ATR evaluator and one for the warfighter. The end product of each framework will be a value score for each ATR system. These scores may then be analyzed for use in decision-making. Figure 3.1 illustrates the approach of translating ATR CS MOPs into a single value score from both perspectives.

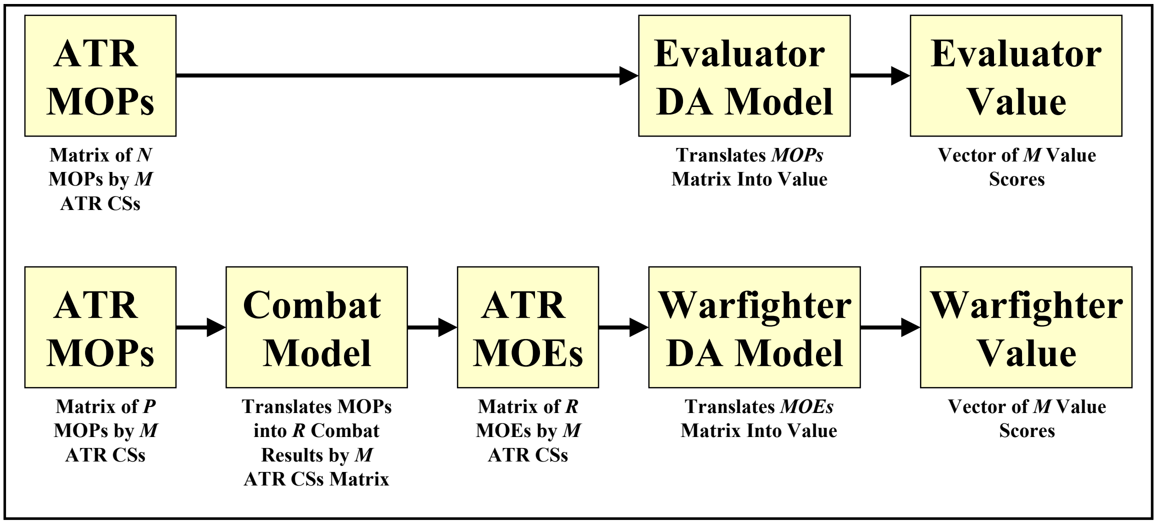

ATR MOPs
Matrix of *N* MOPs by *M* ATR CSs
Evaluator DA Model
Translates *MOPs* Matrix Into Value
Evaluator Value
Vector of *M* Value Scores
ATR MOPs
Matrix of *P* MOPs by *M* ATR CSs
Combat Model
Translates MOPs into *R* Combat Results by *M* ATR CSs Matrix
ATR MOEs
Matrix of *R* MOEs by *M* ATR CSs
Warfighter DA Model
Translates *MOEs* Matrix Into Value
Warfighter Value
Vector of *M* Value Scores

Figure 3.1 Overall MOP Translation Methodology.

The objective of this research is to define an evaluation methodology. Though care was taken with scenario creation and sensor instantiation, the combat models used in this research do have noted limitations.

This chapter is organized as follows. First, an analysis of alternatives approach, as endorsed by the United States Air Force, is reviewed. A summary of the ATR evaluator decision framework construction is given, which includes example results using data from the MSTAR program. Next, a process for translating ATR MOPs into operational MOEs is introduced, followed by an overview of the Extended Air Defense Simulation (EADSIM) combat model. Finally, a description of the warfighters' decision analysis framework is presented.

3.2 Analysis of Alternatives

3.2.1 Overview.

The following section serves as a review of the "basic elements and practices" of analysis, particularly within the United States Air Force (24). Good analytical practices can be standardized to a large extent (24). The following elements of analysis provided the roadmap to the ensuing research and emphasize the importance of tailoring the methodology to meet the goal. Therefore, subsequent use of this ATR evaluation methodology should include a detailed study into the following elements before application.

3.2.2 Goals.

The ultimate goal of ATR technology development is to provide a useful ATR system to the end users. The idea is to develop an ATR system, or a set of ATR systems, that performs better than any other and employ that system in an operational environment. If the technology is well-developed, the ATR system will decrease the time required for a warfighter to make particular decisions by condensing the immense amount of information that must be processed before a decision is made (42). Thus, the real-time battlefield decision process may be shortened, and battlefield management resources may be applied towards other activities (42). The goal must include the impact of *military worth*. Military worth may be summarized through the use of six different attributes: Time to achieve objective, Targets placed at risk, Targets negated (killed), Level of collateral damage, Friendly survivors, and Resources required (24). The alternatives, i.e.

each ATR CS being compared, may then be subjected to a cost versus effectiveness analysis.

3.2.3 Tasks.

Tasks are the means by which goals are achieved. The delineation of these tasks is the responsibility of the decision-maker. These tasks may change as the scenario, i.e. the operational environment, changes. Thus, it is important to either construct a methodology that is robust for all possible scenarios or to clearly state the different scenarios (tasks) with which the results may be associated.

3.2.4 MOEs and MOPs.

The practicality of the DA models described below depends on the accuracy and acceptance of the associated MOPs and MOEs. As defined by the Office of Aerospace Studies, MOEs are used in measuring proficiency in the performance of a task (24). As useful guidelines, MOEs should be associated with a single task, should not be strongly correlated with one another, and should typically be the raw number of an outcome or occurrence. Additionally, cost is never an MOE (24). MOPs are typically "a qualitative measure of a system characteristic chosen to enable calculation of one or more MOEs" (24). Thus, MOPs are used as inputs to describe the system for which the MOEs will be used in the analysis of alternatives.

Most of the MOPs used in this research were established at the time of the MSTAR program. However, MOPs do evolve as the ATR technologies change. New approaches and capabilities to solving target recognition problems require new metrics for assessing ATR performance; thus, MOPs are not necessarily stable. On the other hand, the goals for any given ATR technology should be. Therefore, while a new ATR program may be assessed with a different metric, i.e. the definition of P_D changes, the objective of correctly identifying targets does not.

3.3 ATR Evaluator Decision Analysis (DA) Model

3.3.1 Overview.

The following is a summary of the research captured in the Air Force Institute of Technology technical report entitled “Application of Decision Analysis to Automatic Target Recognition Programmatic Decisions” (40). The research found within the technical report constitutes a joint dissertation research effort and serves as a feasibility study for influencing ATR programmatic decisions using decision analysis (DA) techniques from an ATR evaluator’s perspective (39). The results of the study indicate that DA tools and techniques can be implemented to influence ATR technology programmatic decisions. Figure 3.2 highlights the portion of the performance measure translation methodology being discussed.

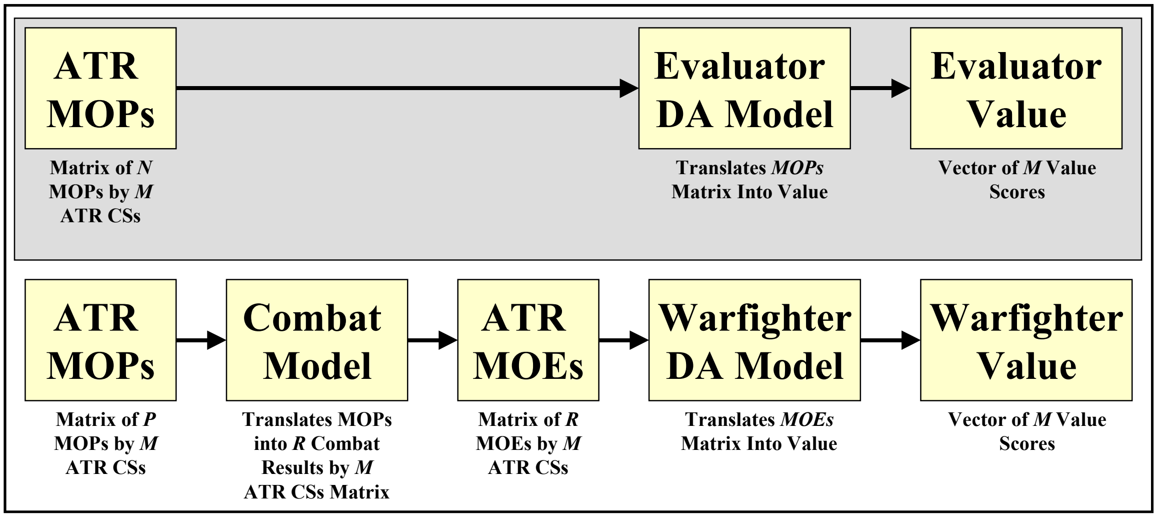

ATR MOPs
Matrix of *N* MOPs by *M* ATR CSs
Evaluator DA Model
Translates *MOPs* Matrix Into Value
Evaluator Value
Vector of *M* Value Scores
ATR MOPs
Matrix of *P* MOPs by *M* ATR CSs
Combat Model
Translates MOPs into *R* Combat Results by *M* ATR CSs Matrix
ATR MOEs
Matrix of *R* MOEs by *M* ATR CSs
Warfighter DA Model
Translates *MOEs* Matrix Into Value
Warfighter Value
Vector of *M* Value Scores

Figure 3.2 Evaluator Portion of the MOP Translation Methodology.

3.3.2 Decision Situation.

To begin construction of the DA framework, a subject matter expert (SME) from AFRL/SN provided input for the construction of the decision framework from an evaluation community perspective. This individual, who represented the various decision-makers and SMEs throughout the ATR evaluation organization, provided all decision-making preferences, values, and utility functions for the model. The SME's value and utility preference structure was elicited over several meetings.

To better understand the construction of the decision analysis model, the different ways in which ATR systems may be employed must be defined. There are two basic employment concepts for ATR technology: Combat Identification (CID) and Intelligence/Surveillance/Reconnaissance (ISR) (40). The CID employment profile is implemented when the primary objective of the ATR CS is to select targets for weapon systems (40). In this scenario, the ATR system is allowed to sacrifice detection performance in order to gain classification accuracy, i.e., selection of a target must be associated with a high degree of confidence as to minimize the number of false alarms. The ISR employment profile is used when the primary objective of the ATR CS is to collect information for many potential targets, i.e., classification accuracy may be sacrificed for improved detection performance (40). For the purposes of the research found in the technical report, each employment concept was considered separately. In this research, the two are combined within a scenario and considered to be of equal importance.

3.3.3 Encoding the Value Hierarchy.

The intention of the research was to create a decision framework that applied to several different ATR technology programs, e.g. MSTAR, AGRI, NCTI, as well as to several different programmatic decision types, e.g. technology investment, transition, or competition. Therefore, a wide variety of performance measures are incorporated into the decision analysis model. Figure 3.2 provides a depiction of the resultant ATR Evaluator DA framework.

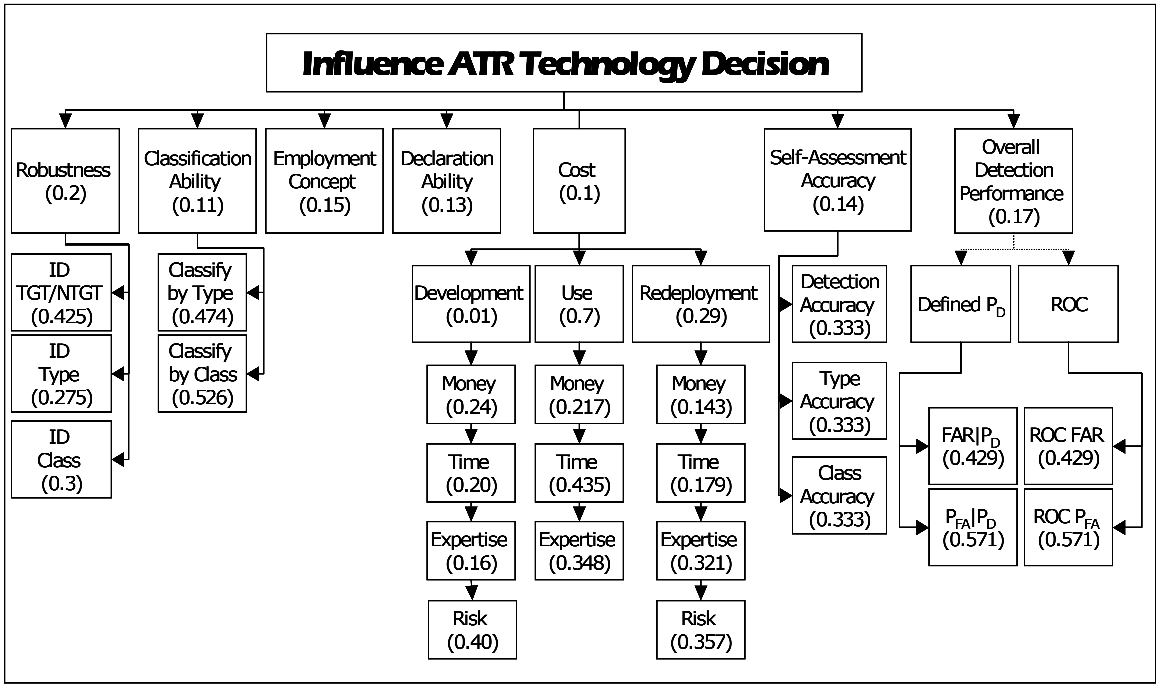


Figure 3.3 Value Hierarchy for Influencing ATR Evaluator Decisions.

The nodes and their associated numbers featured in Figure 3.2 indicate the weight, or value, that the evaluator places on the given objective, or area of performance. For instance, the *Classification Ability* node accounts for 11% of the total influence on the decision. Within the *Classification Ability* performance area, the *Classify by Type* measure of performance constitutes 47.4% of the *Classification Ability* score while the *Classify by Class* measure comprises the remaining 52.6%. The figure also indicates that the evaluator places the most value on the performance measures associated with the *Robustness* objective (20%), while the *Cost* objective is the least important (10%). Table 3.1 gives the total value attributable to each MOP. This total possible value is calculated by multiplying the appropriate weights along each branch of the value hierarchy for a given MOP and represents the total value provided by the MOP were it at its maximum value. This provides insight into the relative importance of each individual MOP to the evaluator. The individual MOPs are also ranked according to this total possible weight in Table 3.1.

Table 3.1 Total Possible Value Attributable to Each MOP (Evaluator).

Objective	MOP	Total Possible Weight	Rank
Robustness	%Δ P_D (TGT/NTGT)	0.0850	4
	%Δ P_{ID} (Type)	0.0550	8
	%Δ P_{CC} (Class)	0.0600	6
Detection Performance	$FAR\|P_D$	0.0729	5
	$P_{FA}\|P_D$	0.0971	3
Employment Concept	Employment Rating	0.1500	1
Declaration Ability	P_{DEC}	0.1300	2
Classification Ability	P_{ID}	0.0521	9
	P_{CC}	0.0579	7
Cost	Development Money	0.0002	21-23
	Development Time	0.0002	21-23
	Development Expertise	0.0002	21-23
	Development Risk	0.0004	20
	Redeployment Money	0.0041	19
	Redeployment Time	0.0052	18
	Redeployment Expertise	0.0093	17
	Redeployment Risk	0.0104	16
	Use Money	0.0152	15
	Use Time	0.0305	13
	Use Expertise	0.0244	14
Self-Assessment Accuracy	$E_{S\text{-}PD}$	0.0466	10-12
	$E_{S\text{-}PCC}$	0.0466	10-12
	$E_{S\text{-}PID}$	0.0466	10-12

The value hierarchy illustrates how ATR system measures of performance influence programmatic decisions. For this example, the weights were elicited for a competition decision between three different ATR CSs. In fact, the framework could be used to influence any number of decisions made by the ATR evaluator, but the corresponding weights on the various nodes would most likely be different (40). This points to the flexibility of the decision analysis approach, but also hints at the time consumption associated with the value elicitation of the decision framework construction. MOPs are introduced to the DA framework to produce a single value score. This score incorporates the preference structure of the decision-maker. Thus, the subjective preferences of the decision-maker are quantified and then translated into a single score, which may be evaluated objectively.

3.3.4 Results Using MSTAR Data.

Since the ATR evaluator data served as a feasibility study, real world data was applied to the model. The performance characteristics of three different ATR systems, labeled A, B, and C, were introduced to the evaluator's model. From discussions with the SME, it was clear that ATR B was considered to be the superior system performance-wise. The following tables and figures provide the results of the data being introduced to the DA framework.

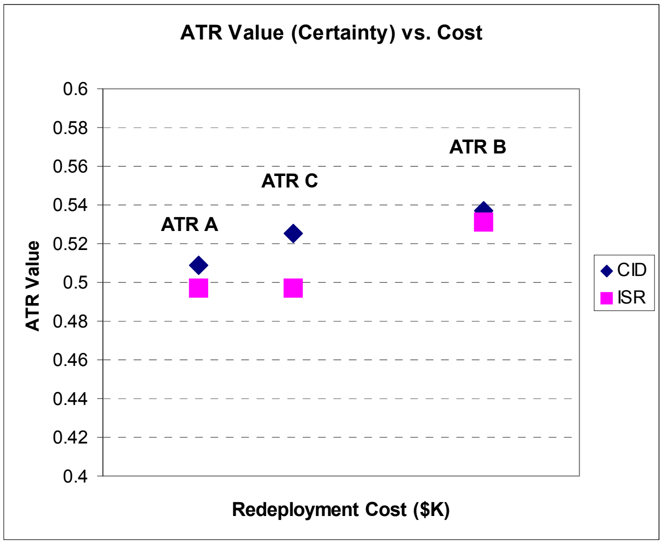


Figure 3.4 Plot of ATR Value Under Certainty.

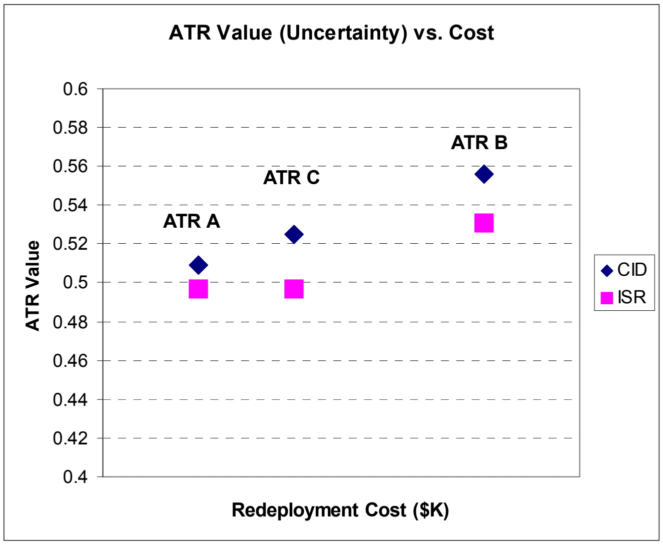


Figure 3.5 Plot of ATR Value Under Uncertainty.

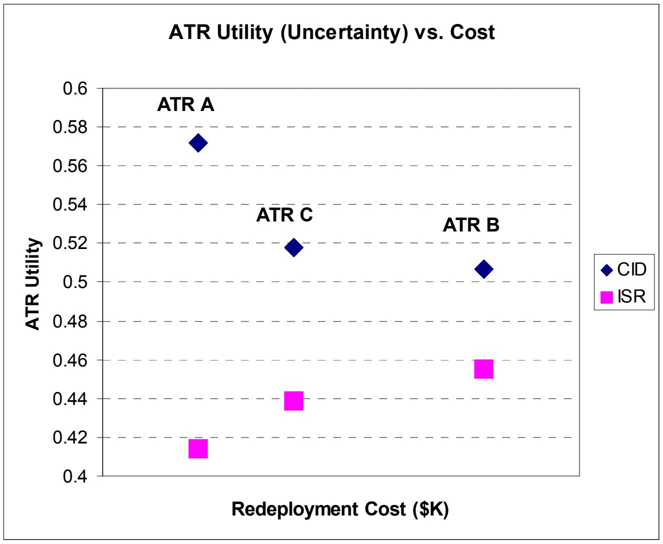


Figure 3.6 Plot of ATR Utility Under Uncertainty.

Analysis of Tables 3.2 and 3.3 would indicate that ATR B is the superior performer in almost every situation. In fact, in terms of value, it ranks as the best ATR CS except when value is assessed for the CID employment concept. In terms of utility, ATR A ranks as possibly a better choice than ATR B for the CID employment concept. In terms of a value-to-cost-ratio, though, ATR C ranks as possibly a better choice than ATR B for the CID employment concept. Since the value score for ATR C lies above an imaginary line between ATR A and ATR B in Figure 3.4, it must have a higher value to cost ratio for the CID case. For the ISR case, it seems that the value and utility measures are much lower. This conclusion follows since the ATR is not expected to perform as well at ISR settings, i.e. a high probability of detection results in a high probability of false alarm. Again, ATR B dominates the field except for ATR C, which again exhibits a higher utility measure and a higher utility-to-cost ratio for the ISR case in Figure 3.6. The conclusion that ATR B is probably the superior ATR CS falls in line with the overall performance assessments given by the ATR evaluators involved in the testing. The results allow a graphical comparison and provide insight into the measurable differences between the ATR CSs.

Table 3.2 ATR CS Expected Value and Expected Utility Results.

		ATR A	ATR B	ATR C
Value (Certainty)	**CID**	0.509	0.537	0.525
	ISR	0.497	0.531	0.497
Value (Uncertainty)	**CID**	0.509	0.556	0.525
	ISR	0.497	0.531	0.497
Utility (Uncertainty)	**CID**	0.572	0.507	0.518
	ISR	0.414	0.455	0.439

Table 3.3 Ranked ATR CS Alternatives by Expected Value and Utility.

		ATR A	ATR B	ATR C
Value (Certainty)	**CID**	3	1	2
	ISR	2	1	2
Value (Uncertainty)	**CID**	3	1	2
	ISR	2	1	2
Utility (Uncertainty)	**CID**	1	3	2
	ISR	3	1	2

3.4 Process for Translating MOPs into MOEs

3.4.1 Overview.

While the ATR evaluator DA framework uses ATR MOPs as a direct input, the warfighter DA model requires MOEs for inputs into the value hierarchy. Thus, two extra steps are required before the warfighter DA model can produce value scores. First, ATR MOPs must be translated into operational results, via a combat model in this case.

Next, the combat model outputs must be slightly altered into the form of MOEs that are of interest to the warfighter. For this research, EADSIM served as the combat model for its ability to accurately model ATR technology effects, its operational modeling level of detail, its acceptance throughout the armed forces simulation community, and its operating system diversity. Figure 3.7 highlights the portion of the performance measure translation methodology being discussed.

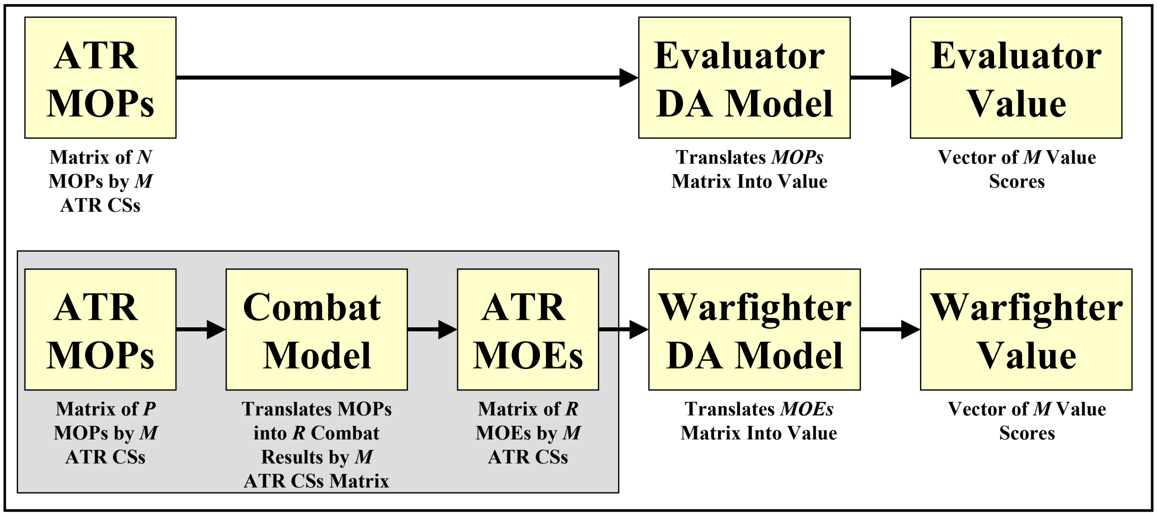

ATR MOPs
Matrix of *N* MOPs by *M* ATR CSs
Evaluator DA Model
Translates *MOPs* Matrix Into Value
Evaluator Value
Vector of *M* Value Scores
ATR MOPs
Matrix of *P* MOPs by *M* ATR CSs
Combat Model
Translates MOPs into *R* Combat Results by *M* ATR CSs Matrix
ATR MOEs
Matrix of *R* MOEs by *M* ATR CSs
Warfighter DA Model
Translates *MOEs* Matrix Into Value
Warfighter Value
Vector of *M* Value Scores

Figure 3.7 Combat Model Portion of the MOP Translation Methodology.

3.4.2 Extended Air Defense Simulation (EADSIM).

EADSIM, produced by Teledyne Brown Engineering, Inc., is the combat model used for producing outputs that may be translated into MOEs for the warfighter DA framework. EADSIM is a system-level simulation designed to be used by combat developers, material developers, and operational commanders to assess the effectiveness of Theater Missile Defense and air defense systems against the full spectrum of extended air defense threats (67). It provides a many-on-many theater-level simulation of air and missile warfare, an integrated simulation tool to support joint and combined force operations, and a tool to augment exercises with realistic air defense training (67). EADSIM allows a user to explicitly model sensors and their interaction with objects within the battle space. EADSIM was selected for its level of engagement detail, hardware platform diversity, and DoD acceptance. For further validity, EADSIM was most recently selected to serve as the constructive model for the Targets Under Trees program (1).

Figure 3.8 depicts the EADSIM data and module architecture, which consists of three separate sections: the Simulation Setup, the Run-Time Models, and the Post-Simulation Analysis (68). For routine use, EADSIM is comprised of three separate modules: Scenario Generation, Scenario Playback, and Post-Processing. The Scenario Generation module allows a user to create a battle space, deploy forces, and select scenario execution preferences. The Scenario Playback module allows the user to view the scenario results in two or three dimensions, highlight objects or actions of interest, and visually debug problems occurring within the execution of a scenario. The Post-Processing module allows the user to select various engagement, detection, and communication statistics for analysis following scenario execution.

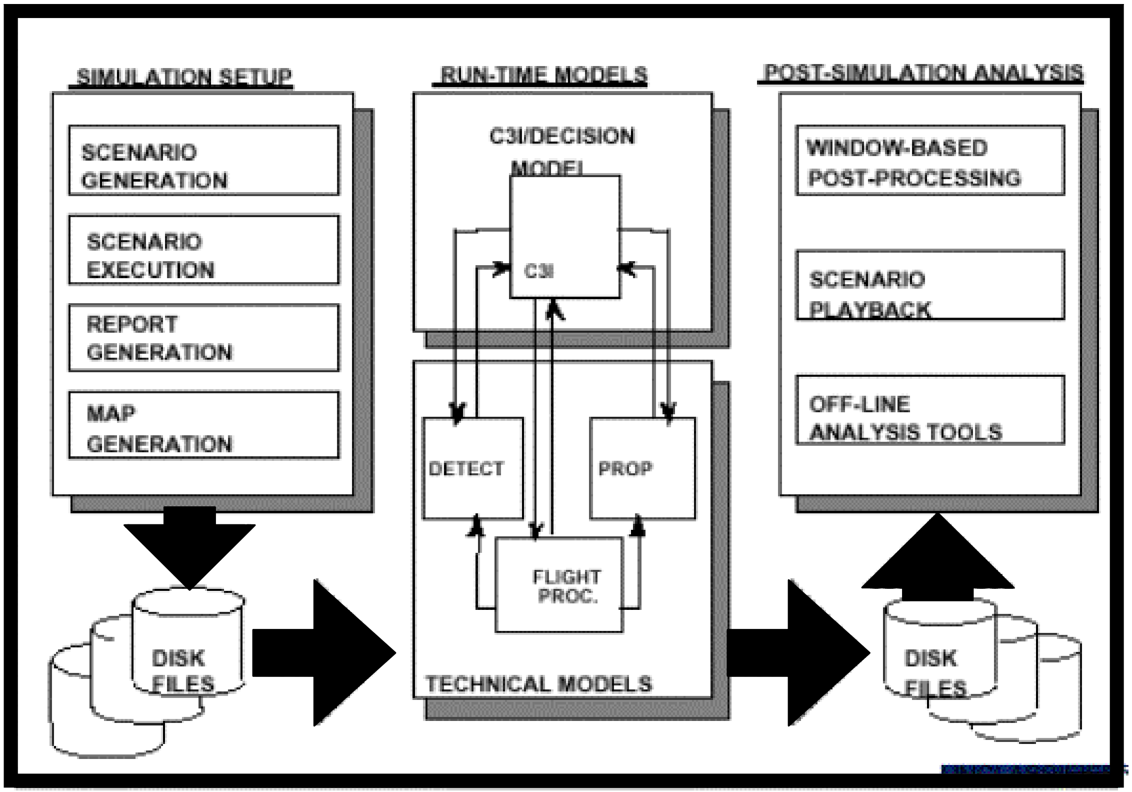


Figure 3.8 EADSIM Architecture (68).

3.4.3 EADSIM Scenario Construction.

EADSIM models battle space players at the system level. An EADSIM user collectively associates various elements along with a ruleset to create a system. Elements used by a particular system include airframes, weapons, communication devices, sensors, formations, and jammers. The ruleset defines how the particular system will react with other systems and actions in the scenario. Once an acceptable system is defined, the user deploys a system as a platform within a scenario. Thus, a single ruleset and applicable elements within EADSIM are grouped together to define a system. A platform is an instance of the particular system deployed within an EADSIM scenario. A user can employ ground-based and airborne platforms easily within an EADSIM scenario.

Additionally, the deployed platforms may be specified as Friendly (Blue force) or Foe (Red force). EADSIM scenarios consist of laydowns, which are specified collections of deployed platforms. Differing platform formations or characteristics may be saved under different laydowns rather than creating an entirely new scenario. Figure 3.9 depicts the relationship between elements, systems, platforms, laydowns, and scenarios in EADSIM.

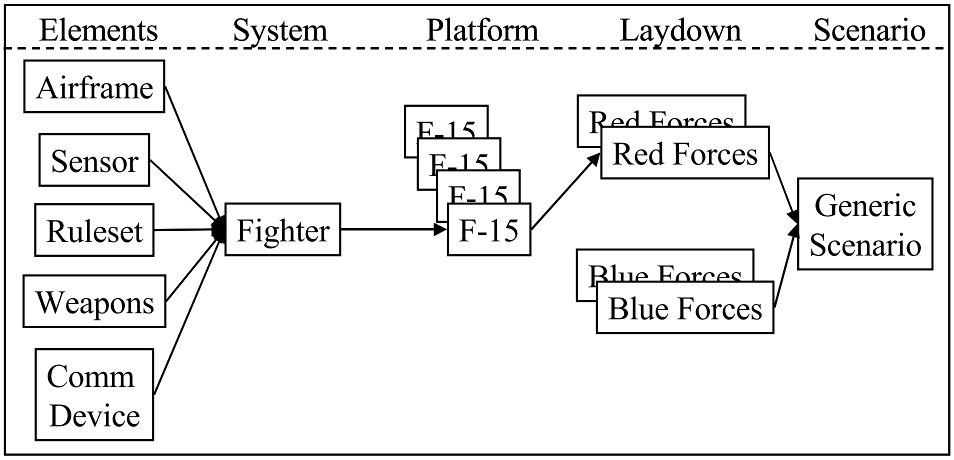


Figure 3.9 Graphical Explanation of EADSIM Scenario Construction.

For the purposes of this research, two different types of airborne platforms were deployed in EADSIM: an ISR platform (using the AWACS ruleset) and a CID platform (using the AG_ATTACKER ruleset). The sensors on each were altered to mimic the performance of a SAR sensor. First, the sensors are pointed at angles and given detection range limitations appropriate for a SAR sensor. The Non-cooperative Target Recognition (NCTR) and Detection Parameters settings were changed at the sensor element level. These changes indicate how well the airborne sensor can detect and classify potential targets of a given type. The Detection Parameters and NCTR settings can be changed at the ground-based component to indicate how effectively the ground-based system hides from the airborne sensor, but were not for this study.

The scenario executes in the following manner. A Blue ISR platform launches from a Blue airbase to detect ground targets in the immediate area, which is roughly 8100 km^2. The ISR platform relays track information to a ground site, which transfers the information to the Ground Commander, or Air Operations Center (AOC). The AOC then schedules air-to-ground fighters, using the ATR technology in the CID mode, for engagement against possible ground targets. The fighters engage only if their own ATR assessment agrees with the ISR platform's assessment at the Friend or Foe level. Thus, if the ISR platform designates a Blue Tank to be hostile, the fighter will only expend a weapon if his own ATR assessment agrees that the target is hostile.

3.4.4 Assumptions.

As for the objects being modeled, a change necessary for EADSIM to accurately model the effects of an ATR CS in the scenario is the removal of Identification Friend or Foe (IFF) devices, which is a reasonable assumption for ground targets. When a new target is detected, it is placed in the track file of the detecting system. One of the first actions by the detecting system is to perform an IFF check. If the IFF is working properly, there is no way for a sensor to misclassify a target. To notice the effects of misclassification, the IFF was disabled on ground platforms. Another correctable issue with EADSIM is the fact that Friendly ground-based units are not added to the track files. Therefore, all Friendly ground-based entities were created as Foes, but tagged as Friends during scenario generation.

There are many different possible scenarios faced by the armed forces of the United States. Whether it is a conventional attack or a peacekeeping mission, the warfighter must be ready. Computation time and the sheer magnitude of possible scenarios prohibit the full study of this area. Thus, a single scenario location and force mixture is used for all of the analysis contained in the research involving EADSIM. There are a few assumptions made throughout the scenario. The first major set of assumptions is the employment of ATR technologies within the scenario. It is assumed that the ISR platform is using a mature ATR technology, orbits the battle space searching for targets, and passes track information to a ground commander for use in vectoring fighters to targets. The fighters serve as the CID platform in the scenario. Once tasked by a ground commander, they must detect the target with their onboard CID sensor before engaging. The ATR technologies on the aircraft are assumed to be the same ATR system operating at different P_D levels. These assumptions, though they disregard additional players within the scenario and do not accurately describe *current* operations, are fairly reasonable. Analysis of the flow of information from the ISR platform to the CID fighters is not the main objective of this research.

Next, within the scenario, engagements are limited to friendly air-to-ground strikes in order to gather battlefield effects based solely on the implementation of ATR systems. It is assumed that enemy aircraft do not attack friendly air forces while they are detecting, classifying, or attacking targets. The underlying assumption is that Friendly forces have achieved air superiority or are providing adequate air protection during sorties. This assumption allows for analysis of the direct effects of employing ATR technology within the operational scenario. If this assumption were not made, it would not be clear if employing ATR technologies enabled the destruction of enemy ground targets.

3.4.5 MOE Translation from Combat Model Outputs.

Executing the EADSIM scenario produces both a playback file and user-selected data files from which post-processing reports may be generated. EADSIM provides the number of systems destroyed, the number of weapons expended, and the number of remaining systems for friendly, hostile and neutral forces. EADSIM also tracks the length of the conflict and records when each detection, engagement, success, and failure occurs. EADSIM post-processing reports may be tailored to capture events or specific platforms of interest.

The combat outputs must be translated for application to the warfighter DA model. Table 3.4 lists the MOEs used by the warfighter DA framework and the EADSIM outputs used to generate them.

Table 3.4 Warfighter DA Model MOEs and Associated EADSIM Outputs.

Objective	MOE	EADSIM Output Utilized
Minimize Hostile Weapons	% of Bombs Left	Engagement Report Analysis
	% of Mass Destruction Left	Engagement Report Analysis
	% of CMs & S/S Left	Success Category
	% of S/A & A/A Left	Engagement Report Analysis
Minimize Hostile Warfighting Systems	% of Systems Left	Success Category
	% of Personnel Left	Engagement Report Analysis
	% of C2 Left	Success Category
Minimize 'Bad Press'	Length of Battle	Determined by Scenario or GA: Scramble
	# of Civilians Killed	Engagement Report Analysis
	# of Civilian Structures Destroyed	Success Category
	# of Fratricide Incidents	Engagement Report Analysis
Maximize Friendly Weapons Remaining	% of Systems Left	Success Category
	% of Personnel Left	Engagement Report Analysis
	% of C2 Left	Success Category
Maximize Friendly Warfighting Systems Remaining	% of Precision Bombs Left	Weapons Category & GA: Scramble
	% of Dumb Bombs Left	Weapons Category & GA: Scramble
	% of CMs & S/S Left	Engagement Report Analysis
	% of S/A & A/A Left	Engagement Report Analysis

The *Length of Battle* MOE is implicitly calculated as a simulation runtime is established during EADSIM scenario creation. Since no other means of halting EADSIM model execution exists, the *Length of Battle* MOE must be calculated in a different, but reasonable, manner or conceded. A possible way to account for the length of time is to run the model for a small segment, say one day, and apply the results of the smaller segment to that of a longer segment, i.e. one week. Another method would be to mark the time when a certain status is reached within the scenario. For instance, the *Length of Battle* time could refer to when the friendly forces destroy at least 50% of the hostile forces. The limitations of this second method are that the result fixes the level of another

MOE (in this case, *% of Hostile Systems Remaining*), and there are no guarantees that the fixed MOE will meet a predefined level throughout the scenario. Finally, this MOE could be held standard across all ATR CSs being evaluated. The method for calculating *Length of Battle* ultimately depends on the preferences of the analyst translating the MOEs and should be explained when presenting the value scores of the warfighter. For the purposes of this study, the *Length of Battle* MOE was assumed to be proportional to the number of sorties accomplished in the given time frame. This method rewards ATR CSs that do not send out fighters against friendly and neutral targets, but penalizes ATR CSs that send many fighters out in the hopes of destroying enemy targets. It is arguable whether or not more sorties increase the length of the battle as more sorties may weaken the enemy's resistance, thereby shortening the war. However, more sorties may also result in more civilian casualties, which may bring more parties into the conflict or cut allied forces involvement.

To calculate the number of personnel and the number of weapons remaining, it is necessary to make assumptions concerning how many are associated with each weapon system. For the given weapon systems in the EADSIM scenario, personnel and weapon amounts were estimated using data from Operation Desert Storm (64). Thus, when a ground target is destroyed, it is assumed that a given amount of ammunition and personnel are destroyed as well.

3.5 Warfighter Decision Analysis (DA) Model

3.5.1 Overview.

The warfighter DA model actually uses a subset of the MOPs utilized by the evaluators' DA model, as seen in Figure 3.10. Thus, no aspect of ATR system cost or risk, as quantified by the ATR evaluator, is included. The result is that the warfighter model ultimately depends solely on the performance characteristics of the ATR CS. Another important difference between the evaluator and the warfighter DA models is that the latter implicitly combines the effects of both employment concepts—ISR and CID, while the former treated them as two different entities. Figure 3.11 highlights the portion of the performance measure translation methodology being discussed.

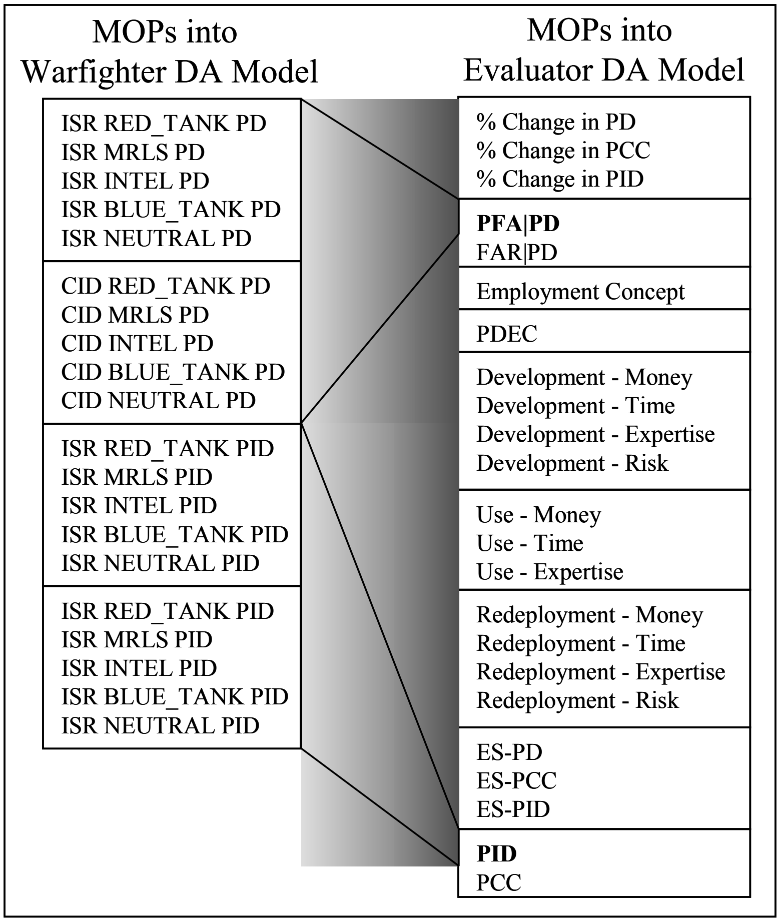


Figure 3.10 Description of MOP Differences Between Evaluator and Warfighter.

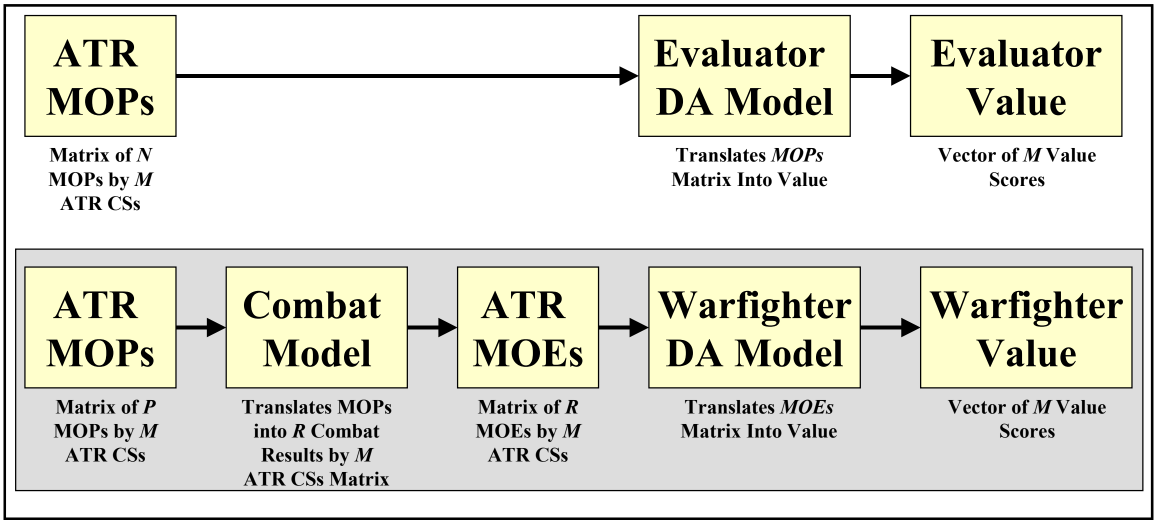

ATR MOPs
Matrix of N MOPs by M ATR CSs
Evaluator DA Model
Translates MOPs Matrix Into Value
Evaluator Value
Vector of M Value Scores
ATR MOPs
Matrix of P MOPs by M ATR CSs
Combat Model
Translates MOPs into R Combat Results by M ATR CSs Matrix
ATR MOEs
Matrix of R MOEs by M ATR CSs
Warfighter DA Model
Translates MOEs Matrix Into Value
Warfighter Value
Vector of M Value Scores

Figure 3.11 Warfighter Portion of the MOP Translation Methodology.

3.5.2 Decision Situation.

For the construction of the warfighter DA framework, an SME from ACC/DRSA refined an original framework developed in preparation for research discussion. The individual defined tasks that directly supported the overall goal of a generic scenario that incorporates ATR technology: achieve the mission objective. The goal is reached with the completion of required tasks defined by the SME. These tasks also incorporate the aspects of military worth previously mentioned. The proficiency of task completion is measured via the associated MOEs. Figure 3.12 illustrates the flow from Goal to Tasks to MOEs.

3.5.3 Encoding the Value Hierarchy.

To begin the warfighter value hierarchy encoding, the SME also provided initial estimates for the value functions (59). Ultimately, three additional individuals from ACC/DR contributed weights used to establish the decision-maker preference structure of the framework. Each individual filled out a sample DA model framework, and the results were averaged to create a collective model (59). The resultant value hierarchy is presented in Figure 3.12.

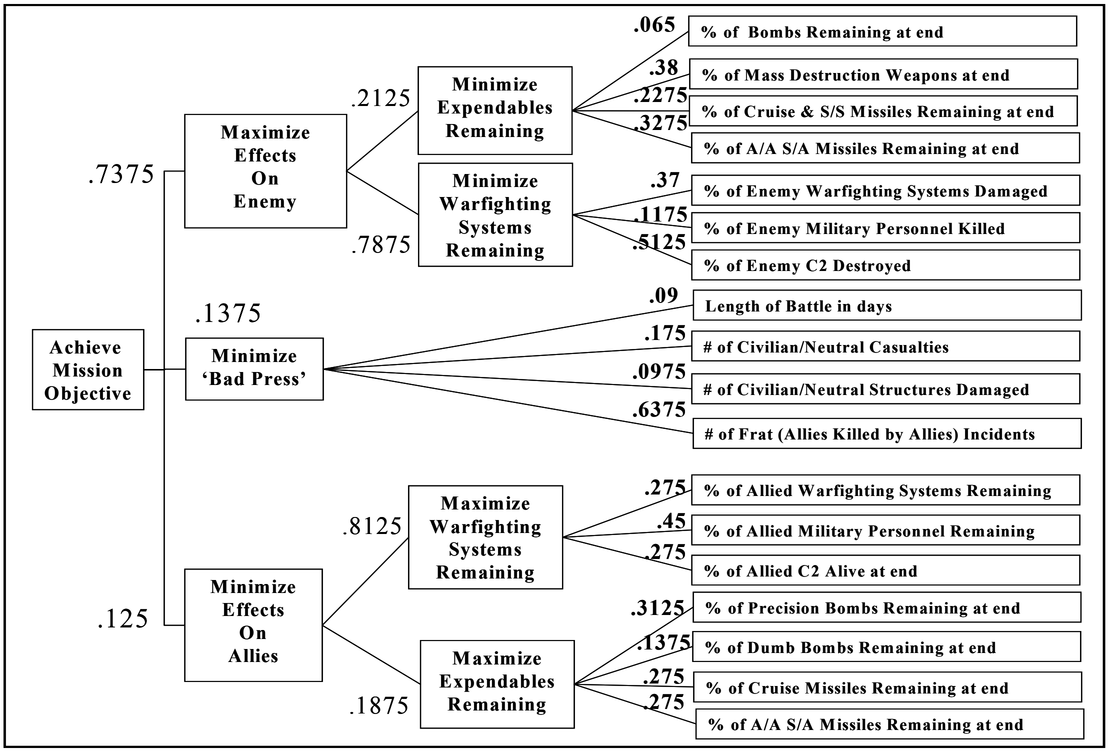

Achieve Mission Objective
.7375
Maximize Effects On Enemy
.2125
Minimize Expendables Remaining
.065
% of Bombs Remaining at end
.38
% of Mass Destruction Weapons at end
.2275
% of Cruise & S/S Missiles Remaining at end
.3275
% of A/A S/A Missiles Remaining at end
.7875
Minimize Warfighting Systems Remaining
.37
% of Enemy Warfighting Systems Damaged
.1175
% of Enemy Military Personnel Killed
.5125
% of Enemy C2 Destroyed
.1375
Minimize 'Bad Press'
.09
Length of Battle in days
.175
# of Civilian/Neutral Casualties
.0975
# of Civilian/Neutral Structures Damaged
.6375
# of Frat (Allies Killed by Allies) Incidents
.125
Minimize Effects On Allies
.8125
Maximize Warfighting Systems Remaining
.275
% of Allied Warfighting Systems Remaining
.45
% of Allied Military Personnel Remaining
.275
% of Allied C2 Alive at end
.1875
Maximize Expendables Remaining
.3125
% of Precision Bombs Remaining at end
.1375
% of Dumb Bombs Remaining at end
.275
% of Cruise Missiles Remaining at end
.275
% of A/A S/A Missiles Remaining at end

Figure 3.12 Warfighter DA Framework.

In regards to military worth, this DA model includes all of the aforementioned items of interest. The *Length of Battle* MOE refers to the time required to achieve the objective. The MOEs that involve civilian deaths, destroyed civilian structures, and allied fratricide attempt to place a value on the level of collateral damage caused by fighting. The *Maximize Warfighting Systems Remaining* objective highlights the importance of targets placed at risk and the value added by having friendly survivors. The *Minimize Warfighting Systems Remaining* objective captures the military worth of killing enemy targets. Finally, the *Maximize Expendables Remaining* objective attempts to capture the value behind the resources required to fulfill the objective.

Typically, inputs into a value hierarchy are of the same units, e.g. dollars, time, etc. However, in this case, the various MOEs do not share the same unit structure. For instance, Length of Battle is measured in days while the majority of the MOEs are percentage measures. This situation requires a value (utility) function to translate the raw MOEs into a space that makes the values compatible throughout the DA model. The primary function of the value (utility) functions is to force the MOEs onto a 0 to 1 value (utility) scale for comparison sake. For this process, the SME served as the decision-maker for constructing value and utility functions for the MOE inputs (59). Utility functions were based upon reaching an 80% solution relative to the value functions (59). Figures 3.13 and 3.14 illustrate the various value and utility curves employed within the warfighter's DA model. Due to the similarity in most of the value and utility functions, there will be a similarity in value and utility scores.

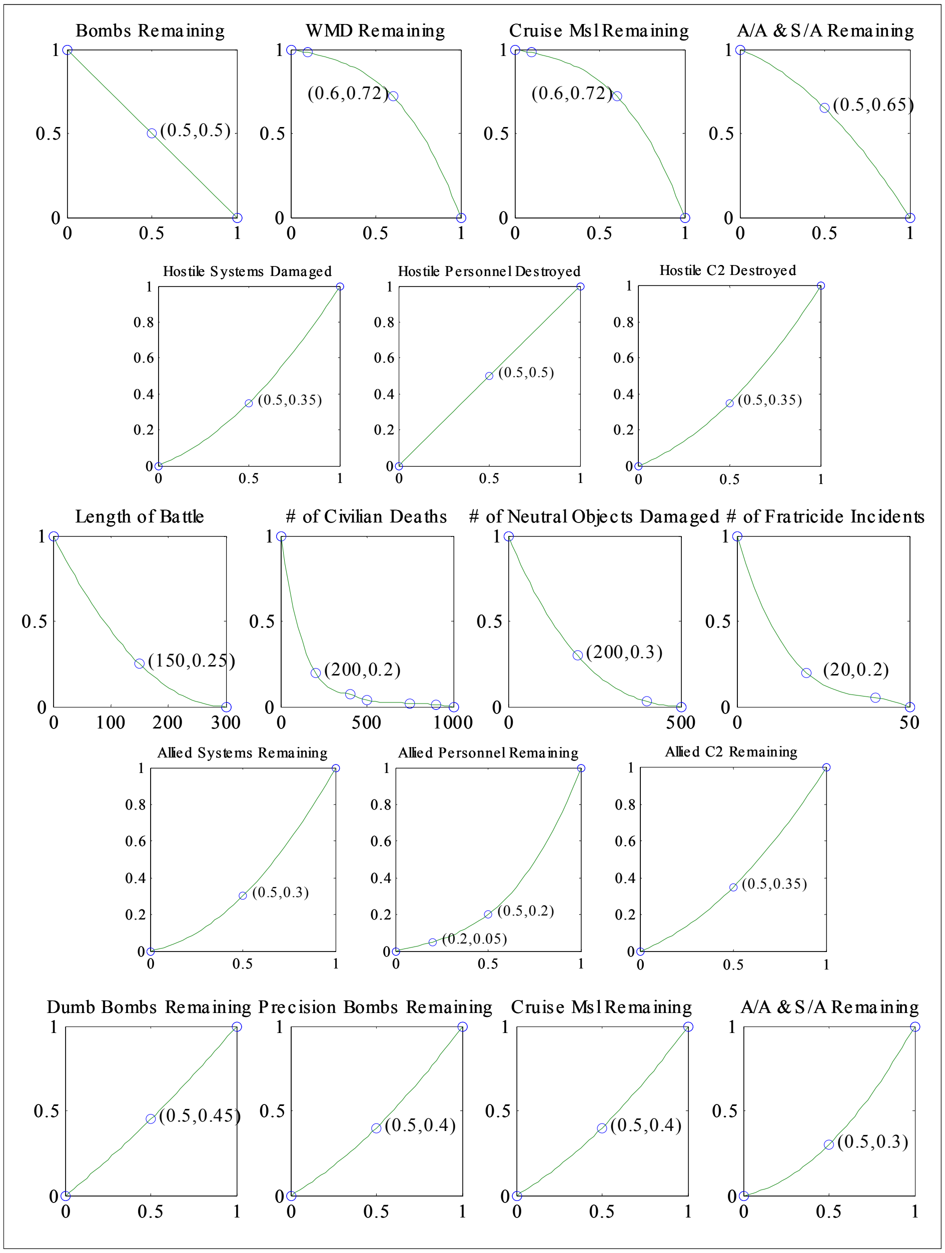


Figure 3.13 Value Functions for Warfighter DA Model.

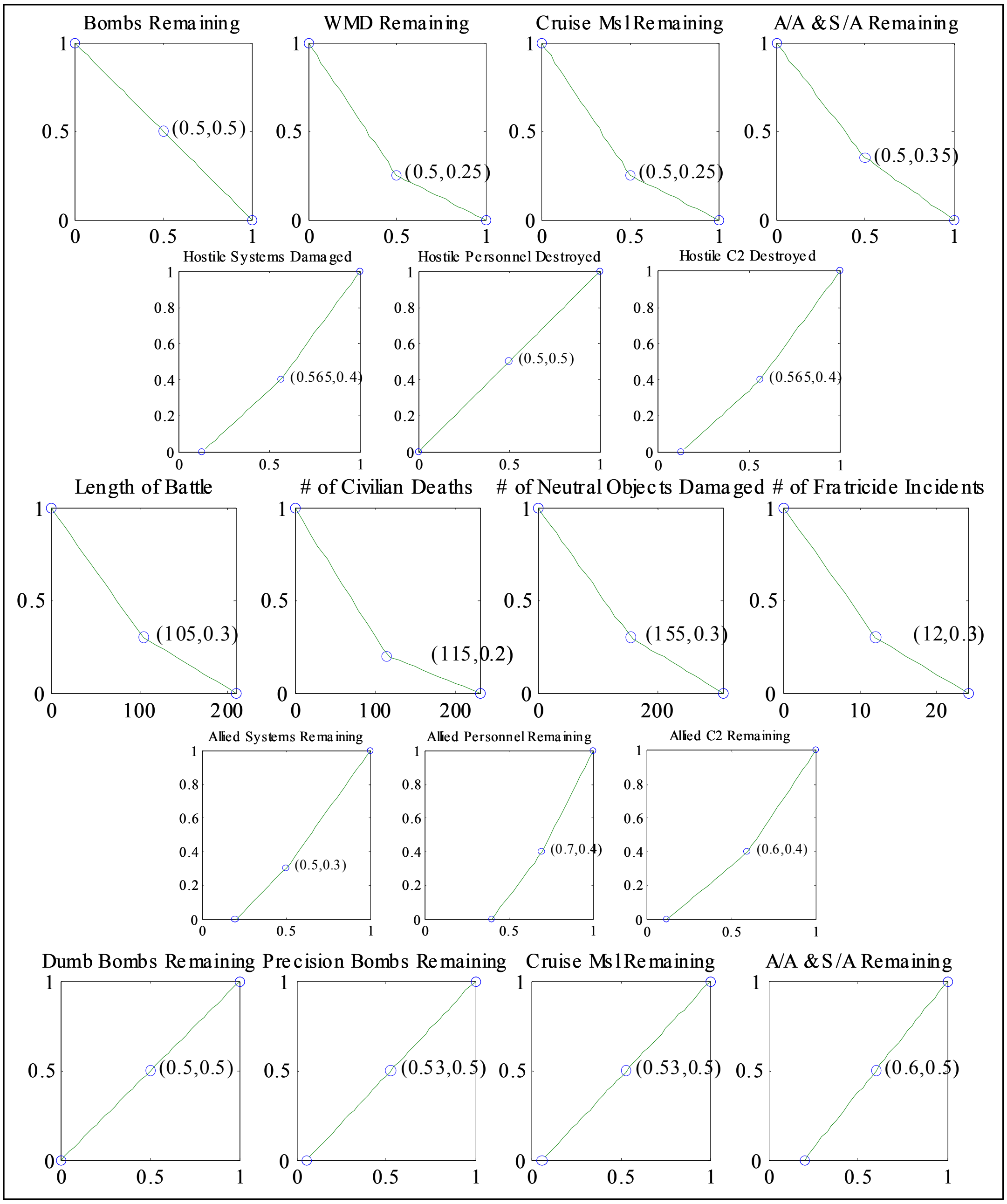


Figure 3.14 Utility Functions for Warfighter DA Model.

Table 3.5 provides the total possible value attributable to each MOP. The total possible value attributable is calculated by multiplying the appropriate weights along the value hierarchy for each individual MOE. The total possible value represents the amount of value the MOE would contribute to the overall score were it at its maximum value. The MOEs are also ranked by their respective total possible weights in Table 3.5. The results provide insight into the importance of each individual MOE.

Table 3.5 Total Possible Value Attributable to Each MOE (Warfighter).

Objective	MOE	Total Possible Value	Rank
Minimize Hostile Weapons	% of Bombs Left	0.0102	14
	% of Mass Destruction Left	0.0596	5
	% of CMs & S/S Left	0.0357	8
	% of S/A & A/A Left	0.0513	6
Minimize Hostile Warfighting Systems	% of Systems Left	0.2149	2
	% of Personnel Left	0.0682	4
	% of C2 Left	0.2977	1
Minimize 'Bad Press'	Length of Battle	0.0124	13
	# of Civilians Killed	0.0241	11
	# of Civilian Structures Destroyed	0.0134	12
	# of Fratricide Incidents	0.0877	3
Maximize Friendly Weapons Remaining	% of Systems Left	0.0279	9-10
	% of Personnel Left	0.0457	7
	% of C2 Left	0.0279	9-10
Maximize Friendly Warfighting Systems Remaining	% of Dumb Bombs Left	0.0032	18
	% of Precision Bombs Left	0.0073	15
	% of CMs & S/S Left	0.0064	16-17
	% of S/A & A/A Left	0.0064	16-17

IV. Application

4.1 Overview

This chapter details how the proposed evaluation methodology could be applied to mature ATR technologies for influencing programmatic decisions or providing combat model analysis involving ATR technologies. First, a combat scenario and performance characteristics for three different ATR systems are generated. These MOPs are then applied to the ATR evaluator DA framework and produce a single value score for each ATR system. Next, the performance characteristics are introduced to EADSIM, the combat model. EADSIM results are then translated into MOEs. The MOEs are then applied to the warfighter DA framework to produce a value score for each system. Finally, the two different value scores are analyzed within the decision analysis context.

4.2 Scenario and Measures of Performance (MOPs) Generation

The following scenario considers three different, mature ATR systems for evaluation. The scenario involves an airborne ISR platform, multiple CID air-to-ground fighters, and various friendly, enemy, and neutral (FEN) ground targets. The purpose of the airborne friendly units within the scenario is to detect all FEN targets while accurately distinguishing between those to be destroyed (enemy) and those to be avoided (friendly and neutral). The objectives of the ISR platform are to detect as many potential targets as possible and relay the tracking information to the ground commander, who may vector air-to-ground fighters to the potential targets. The objectives of the CID platforms include responding to potential targets handed out by the ground commander, detecting the targets using its own sensor, and destroying the target if the original ISR classification is verified.

The definitions of the ISR and CID employment settings, summarized in Table 4.1, are based upon recommendations held by various ATR decision-making authorities. The operating areas of the two different employment settings are marked along notional ROC curves in Figures 4.1 and 4.2. To describe the process conceptually, ATR A will operate at an ISR and a CID level, each having an associated P_D and P_{FA} performance level that produces a ROC curve (Figure 4.1). With a given P_D setting, a P_{ID} and P_{FID} level may be generated from a point on the P_{ID} ROC curve (Figure 4.2). The definitions for P_D, P_{FA}, and P_{ID} presented in Section 2.2.3 are used. The probability of false identification term, P_{FID}, may be thought of as a false alarm measurement where the false alarms consist of all other non-type targets instead of clutter objects, or

$$P_{FID-Type} = \sum_{All\ Non-Types} \mathrm{P(D,"Type"|\ Non\text{ - }Type)}, \tag{4.1}$$

such that

$$P_{FID-Type} = \sum_{All\ Non-Types} \mathrm{P(``Type"\ |\ D, Non\text{ - }Type)P(D\ |\ Non\text{ - }Type)} \tag{4.2}$$

where D is the event of a target detection, and “Type” is the declaration of an ROI as a “Type” target. In other words, when focusing on $P_{ID\text{-}MRLS}$, $P_{FID\text{-}MRLS}$ accounts for all Red_Tanks, Intel Trucks, Blue_Tanks, and Neutral trucks rather than clutter objects.

For this study, the overall P_D and P_{ID} MOPs are decomposed into an individual measure for each of the five target types, i.e. $P_{D\text{-}Type}$ and $P_{ID\text{-}Type}$. For example,

$$P_{ID\text{-}MRLS} = \mathrm{P(D,"MRLS"|MRLS)}, \tag{4.3}$$

where D is the event of a target detection, and "MRLS" is the declaration of an ROI as a MRLS. Hence,

$$P_{ID\text{-}MRLS} = \text{P}(\text{“MRLS”}|\text{D,MRLS})\text{P}(\text{D}|\text{MRLS}). \qquad (4.4)$$

The first term to the right of the equality is taken from the approximate *identification* ROC curve. The second term is taken from the appropriate *detection* ROC curve. In this study, a constant probability of detection is assumed for all target types. This method allows a higher input resolution when introducing the MOPs into a combat model.

Table 4.1 ISR and CID Operational Definitions (5).

Employment Settings	Probability of Detection	Probability of Identification	Probability of False Alarm
ISR	$P_D > 0.9$	$P_{ID} > 0.7$	Large
CID	$P_D > 0.6$	$P_{ID} > 0.95$	Small

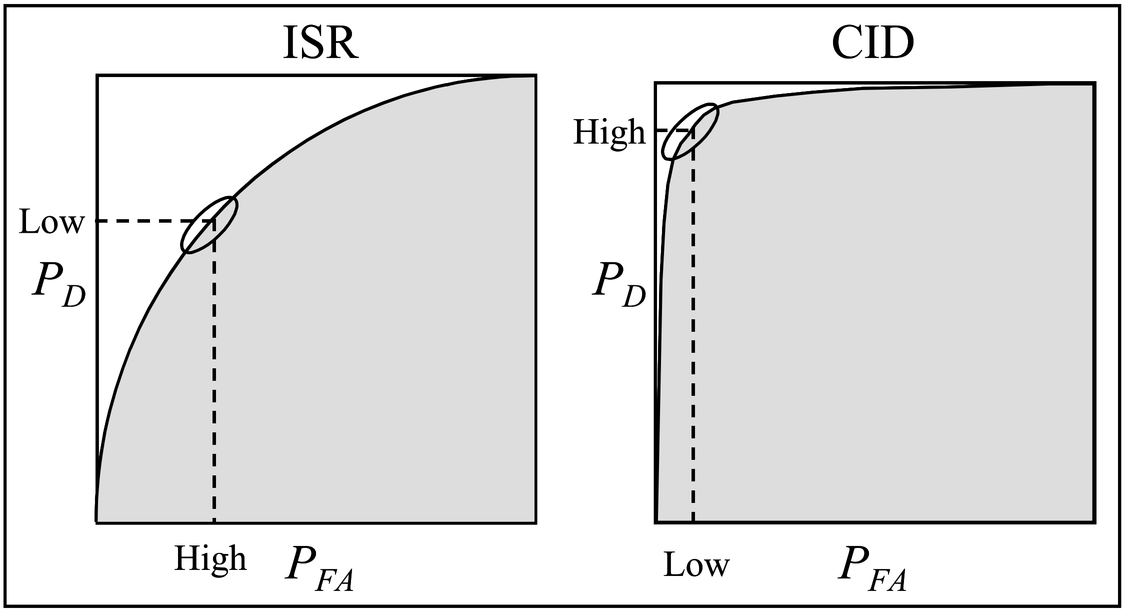


Figure 4.1 Detection Performance Regions of ISR and CID Employment Settings.

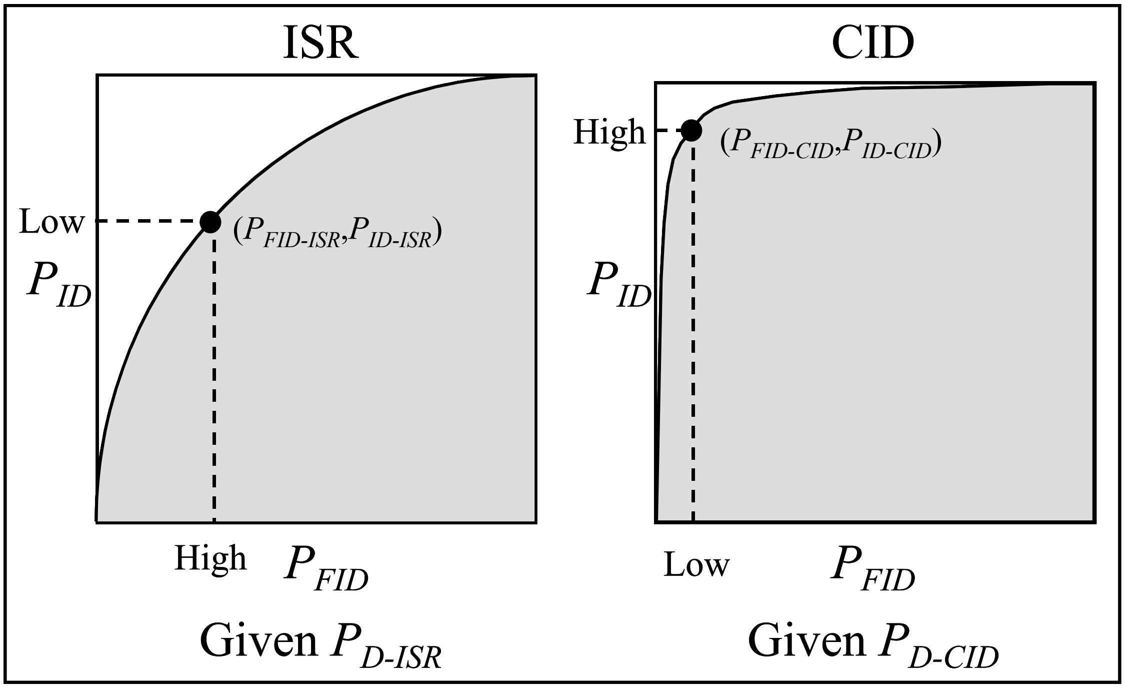


Figure 4.2 Identification Performance Points of ISR and CID Employment Settings.

The ATR systems, labeled A, B, and C, have differing performance advantages and disadvantages. The performance measures for each ATR system are generated using a Microsoft Excel® worksheet. First, Gaussian target and non-target population densities are chosen to represent the possible objects to be found within the scene. As described in Section 2.2.7, a distance measure, Δm, represents the difference between the means of the two populations. A spread ratio, *s*, represents the ratio of the standard deviation of the target population to the non-target population. For generating MOPs to be used in this study, the spread ratios are all assumed to be one as the force mixture of the scenario would be unknown to the ATR developers. These two measures are then used to construct continuous, binormal ROC curves and the associated performance data. For example, using the performance data for ATR A and given an ISR $P_{D\text{-}RED\_TANK}$ of 0.9 (the P_D used for all target types in the ISR setting), the population density measures for the

Red Tank population, which corresponds to an enemy tank known to each ATR system, compared to other non-Red Tanks type targets has a distance measure of 1.43 and a spread ratio of 1.0, as seen in Table A.2. The two measures correspond to $P_{ID\text{-}RED\_TANK}$ settings of 0.890 and 0.830 and $P_{FID\text{-}RED\_TANK}$ of 0.123 and 0.318 when the $P_{D\text{-}RED\_TANK}$ is set to 0.6 (CID) and 0.9 (ISR), respectively. The data for each ATR and potential target can be found in Tables A.1 and A.2. Figures 4.3, 4.4, and 4.5 depict the ISR and CID ROC curves for ATR systems A, B, and C. Using the average *AUC* measure in Table A.2 to compare between the systems, it is unclear which is the best system as ATR A is the best CID system, while ATR B is the best ISR system. It would seem that ATR C is the better performer at a CID setting while ATR B is the better performer when operating in an ISR mode when examining the overall averages of the P_{ID}, P_{CC}, and P_{FA} measures of the two different employment settings in Table A.1. However, it is meaningless to examine the measures outside of their P_D-P_{FA} and P_{ID}-P_{FID} relationships. For example, an ATR CS could have a P_{ID} of 0.999, but it may occur when P_{FID} also equals 0.999, which is a very undesirable performance level. Also notice that the ISR and CID levels are set at the most liberal levels, i.e. P_D is set to 0.6 and 0.9. This provides a worst-case scenario for the combat model results in terms of P_D.

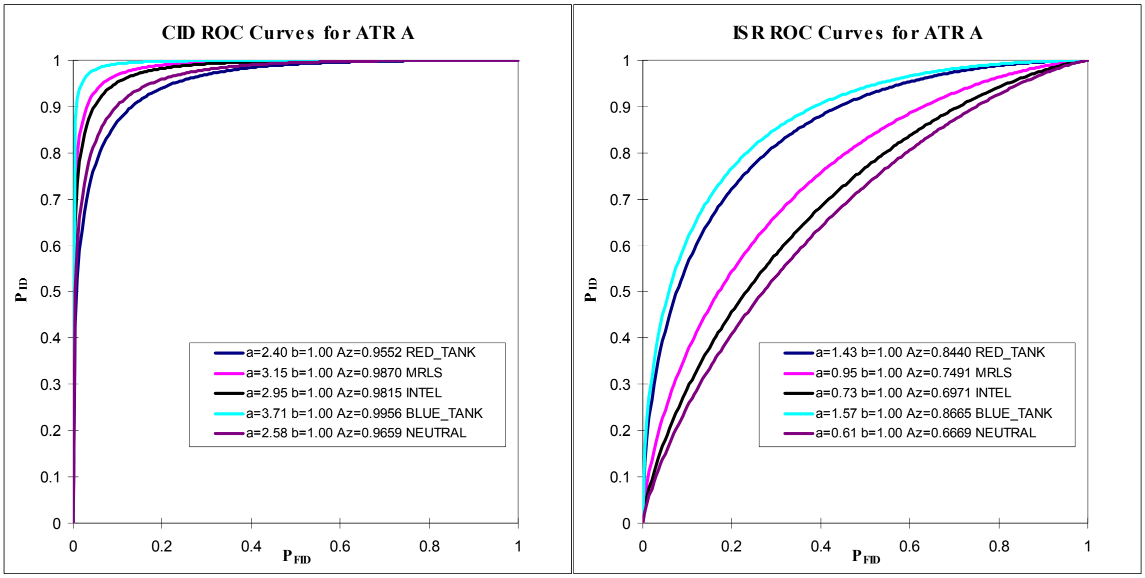


Figure 4.3 ATR System A Performance Expressed Through ROC Curves.

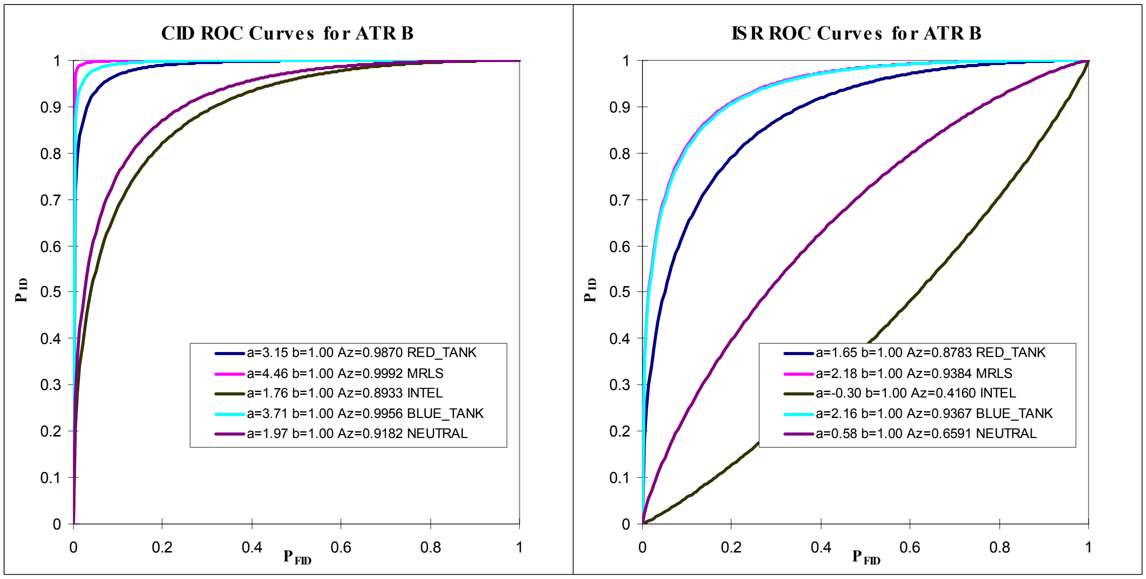


Figure 4.4 ATR System B Performance Expressed Through ROC Curves.

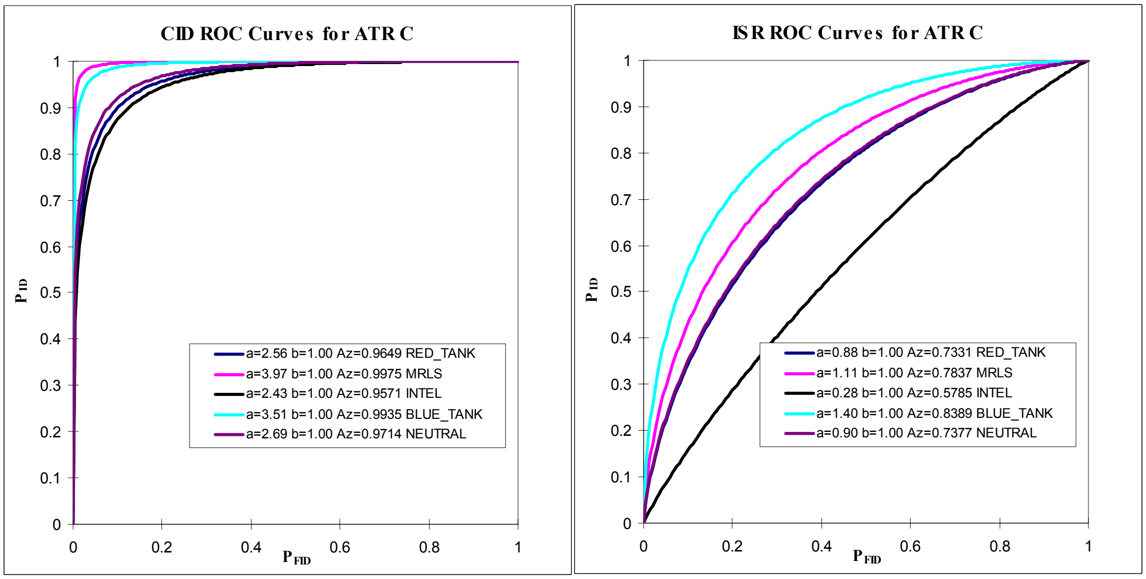


Figure 4.5 ATR System C Performance Expressed Through ROC Curves.

Each of the three notional ATR CSs also has an associated probability of false alarm and false alarm rate, as seen in Tables A.1 and A.3. The *FAR* is different from the P_{FA} in that it refers to the amount of clutter items, i.e., natural objects, buildings, and untrained vehicles, which will be detected and perhaps classified as targets when a given area is scanned. The P_{FA} measure refers to the probability that a non-target will be incorrectly classified as a target. To incorporate *FAR* and P_{FA} measures for a particular sensor within EADSIM, intensive time requirements and advanced sensor modeling are necessary. Due to these difficulties, *FAR* and P_{FA} effects were left out of EADSIM and the surrogate model as well. Leaving the false alarm performance out of the combat models ignores the effects of CID platforms being tasked to seek and possibly destroy civilian structures, vehicles, and natural objects not included in the trained target set. These effects predominantly impact the number of sorties generated and weapons used (as the number of detected objects increases), the number of neutral entities destroyed (as more clutter objects are detected), and the length of the battle (as there should be more total objects to evaluate and the proportion of detected enemy targets to the total number of detected targets shrinks). Thus, for this study, no clutter, i.e. untrained, objects are considered within the combat models. Instead, the P_{FID}, which refers to simply *non-type* objects declared a different type rather than *unknown* objects declared as a certain type, is used within the combat models rather than the traditional P_{FA}. For instance, the $P_{FID\text{-}MRLS}$ measure considers the other four objects of interest (Red_Tank, Intel, Blue_Tank, and Neutral) to be non-targets. This concept is visualized in Figure 4.6 using the MRLS population as an example.

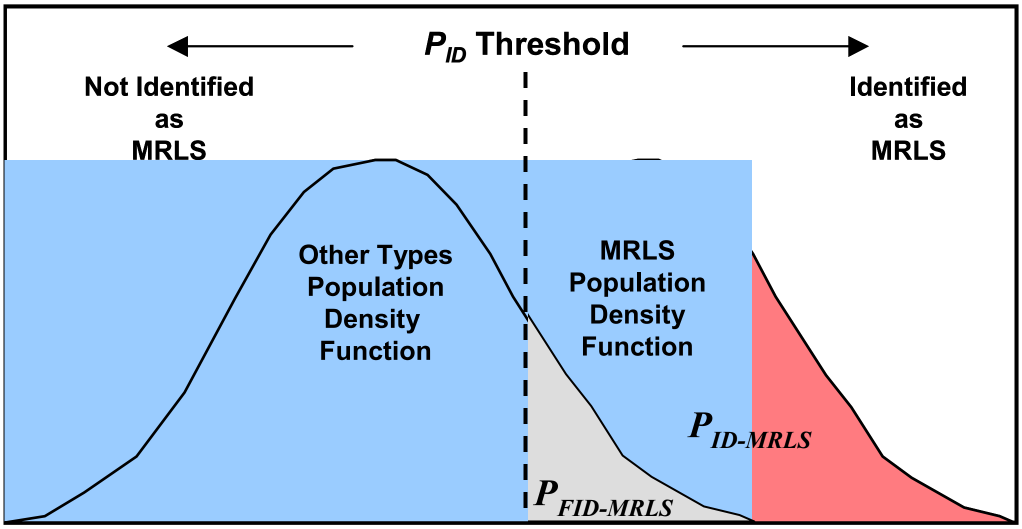


Figure 4.6 P_{ID} and P_{FID} Calculation Concept for EADSIM Input.

Though not used within the combat models, the *FAR* of each ATR CS is used as an input to the Evaluator DA model. The false alarm rates, which are optimistic, were generated for this scenario to negate the false alarm effect that might be experienced via combat model execution. Therefore, the *FAR*s may be realistic under a few assumptions.

First, the ATR systems are assumed to be mature. The false alarm performance of future ATR CSs is uncertain, but should always improve as ATR technology improves. Therefore, it is reasonable to assume that a mature ATR system may have such a FAR in the future. Next, the map used for the scenario is focused around a desert. In fact, a Kuwaiti map is used as the backdrop for the scenario. In such an environment, the number of false alarms should be drastically reduced due to a low incidence of foliage and a low number of vehicles in the area. Finally, it could be assumed that the ISR platform excludes detections outside a certain area of interest, i.e. the battlespace, which could dramatically reduce the impact of the *FAR*.

As an MOP input to EADSIM, P_{ID} is considered to be a given performance measure in response to a given P_D level. Thus, the P_{ID}, associated with a given P_D and P_{FA}, has been estimated by an ATR evaluator through testing. The P_{ID} measure is pivotal when introducing ATR CS MOPs into EADSIM. ATR performance measures are introduced into EADSIM in the following way. First, a P_D is defined for every target object within the scenario. Additionally, a P_D may be defined for each of the target types within EADSIM, as it was in this study. Next, for each target type, a Non-Cooperative Target Recognition (NCTR) matrix is defined. This matrix, which sums to one, details the probability that EADSIM is to declare a particular object (in truth) should it detect it within the area scanned by the radar system. Using the example given in Table 4.2, the matrix represents the identification options for a Red_Tank. Thus, if a sensor detects an object (known to be a Red_Tank object by EADSIM), the sensor has an 83% chance of correctly identifying the object as a Red_Tank. Similarly, the sensor has a 17.7% chance of declaring the object as a Blue_Tank. The P_B, P_R, and P_U columns correspond to the probabilities associated with declaring a target type as *Blue* (Friendly), *Red* (Enemy), or *Unknown*. In this study, the target types are automatically associated with their correct FEN association to ensure that all Red targets are attacked, which is reasonable under the assumption that Red_Tanks are of a certain type, e.g. T-72, and Blue_Tanks are from another, e.g. M-1A1. For example, if a tank-like object is detected and classified as a T-72, there is no reason to conclude that the tank is friendly. Thus, the probability of classifying an object identified as a Red_Tank as a Blue (Friendly) object is zero, which is the first entry in the matrix.

Table 4.2 Example EADSIM NCTR Matrix for a Red_Tank Object.

		To Be Classified As		
	Target Type	P_B	P_R	P_U
To Be Identified As	**RED_TANK**	0.00	0.830	0.00
	MRLS	0.00	0.048	0.00
	INTEL	0.00	0.024	0.00
	BLUE_TANK	0.177	0.00	0.00
	NEUTRAL	0.027	0.00	0.00

The robustness measures for the ATR CSs were randomly generated. A random percentage value between five and twenty constitutes the degradation that the ATR CS encounters when moving from the ideal target profile to an altered one, i.e. from the nominal (trained) setting to a different (untrained) setting. The assumption is also made that the Self-assessment capability is not used on these ATR systems. Finally, the cost and employment concept data for each system was based loosely upon the data and assumptions used for evaluating the data from the MSTAR program and are listed in Table A.3.

4.3 Evaluator DA Framework Application

The data from Table A.1 is introduced into the evaluator DA framework. The results in Table 4.3 indicate that the preference structure elicited from the evaluator regards ATR C as the best ATR of the three. Table 4.3 provides the rankings for the evaluator. Figures 4.7 and 4.8 depict the value and utility scores for the ATR evaluator. Notice that the evaluator's value scores are given in both ISR and CID mode as well as a total value. The total value is merely the average of both the ISR and CID values and will allow subsequent comparison to a warfighter value score.

Table 4.3 ATR Evaluator DA Model Results (Bold indicates highest score).

	Employment Mode	**ATR A (Rank)**	**ATR B (Rank)**	**ATR C (Rank)**
Value	**CID**	0.5913 (3)	0.5982 (2)	**0.6282 (1)**
	ISR	0.5798 (3)	0.5994 (2)	**0.6238 (1)**
	TOTAL	0.5856 (3)	0.5988 (2)	**0.6260 (1)**
Utility	**CID**	0.5135 (2)	0.5071 (3)	**0.5769 (1)**
	ISR	0.4484 (3)	0.4589 (2)	**0.5018 (1)**
	TOTAL	0.4809 (3)	0.4830 (2)	**0.5394 (1)**

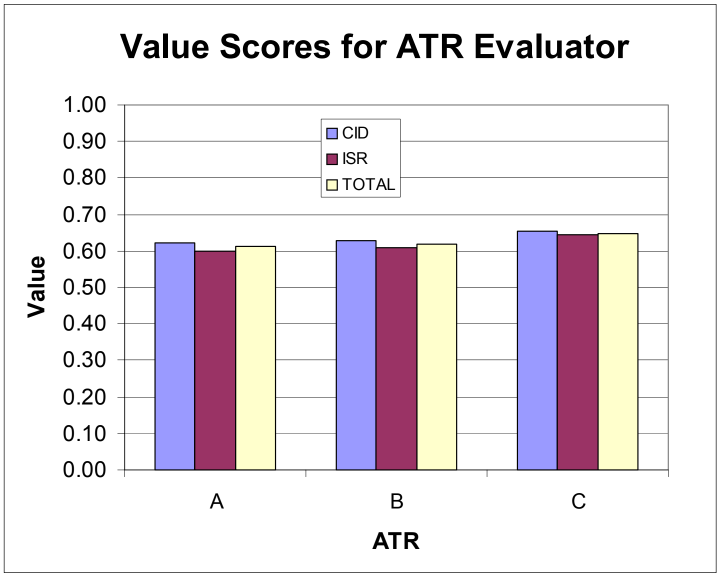


Figure 4.7 Evaluator Value Scores.

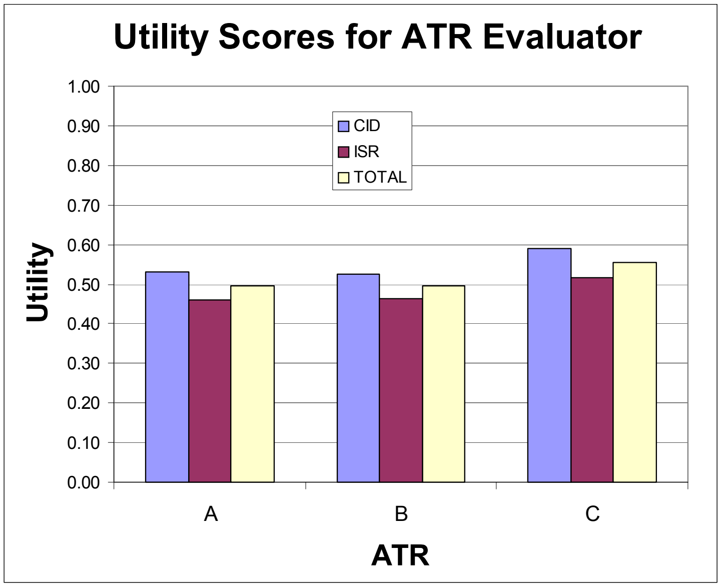


Figure 4.8 Evaluator Utility Scores.

4.4 Warfighter DA Framework Application

To assess the value and utility of the warfighter perspective, the MOPs that strictly detail ATR performance, which includes the probabilities of detection, identification, classification, and false alarm but excludes costs, risks, and employment concept information, is then introduced to EADSIM. Thus, the MOPs used in EADSIM focus solely on the measures that assess actual ATR operational performance. The combat results for four different EADSIM runs are summarized in Table 4.4. The MOEs are introduced into the warfighter DA model. Thus, the raw MOEs are translated via a value function and incorporated into the warfighter DA value hierarchy. The results listed in Table 4.5 indicate that the warfighter would prefer ATR B to the other two alternatives. Figure 4.9 illustrates the value and utility attributable to each of the warfighter's goals. Figure 4.10 depicts the *amount* of value each objective composes of the total value score, while Figure 4.11 depicts the *actual percentage* of value each objective contributes compared to the total possible value contributable by each objective. From Figure 4.10, it is evident that operations that impacted the effect on the enemy contributed the most value associated with each ATR CS (largest shaded region for each ATR), as would be expected with the highest valued objective (Figure 3.12). To compute the objective percentage pictured in Figure 4.11, the value of each objective is divided by the appropriate top-level weight of the warfighter value hierarchy pictured in Figure 3.12. This graph provides insight into which objective is closest to being perfect in its contribution towards the overall value. In this case, ATR C is the best performer in terms of the *Minimize Bad Press* and *Minimize Effect on Allies* objectives, while ATR B is the best performer in the *Maximizing Effect on Enemy* objective. Since the latter is the highest valued objective, it is evident why ATR B scores so highly in terms of value.

Table 4.4 MOE Averages from EADSIM (4 Repetitions).

MOE	ATR A	ATR B	ATR C
Pct of Dumb Bombs Remaining	1	1	1
Pct of WMD Remaining	1	1	1
Pct of S/S Missiles Remaining	0.3571	0.3929	0.4286
Pct of A/A & S/A Remaining	1	1	1
Pct Red Forces Damaged	0.4327	0.5577	0.4327
Pct of Red Personnel Killed	0.3250	0.3173	0.3615
Pct Red C2 Damaged	0.3750	0.6250	0.3125
Length of Battle	18.7500	30.7500	17.2500
Number of Dead Civilians	1.7500	2.7500	1.2500
Number of Damaged Neutrals	0.7500	1.0000	0.5000
# of Fratricide occurrences	3.7500	3.2500	0.0000
Pct Blue Weapons Remaining	0.9375	0.9375	1
Pct of Blue Personnel Remaining	0.9375	0.9458	1
Pct of Blue C2 Remaining	1	1	1
Pct Remaining Dumb Bombs	0.9846	0.9790	0.9841
Pct Remaining Precision Bombs	0.9840	0.9775	0.9859
Pct Remaining CMs	0	0	0
Pct Remaining S/A	0	0	0

Table 4.5 Warfighter Value and Utility Scores (Bold indicates highest score).

	ATR A	ATR B	ATR C
Value	0.3988	**0.5061**	0.4140
Utility	0.3845	**0.4908**	0.3923

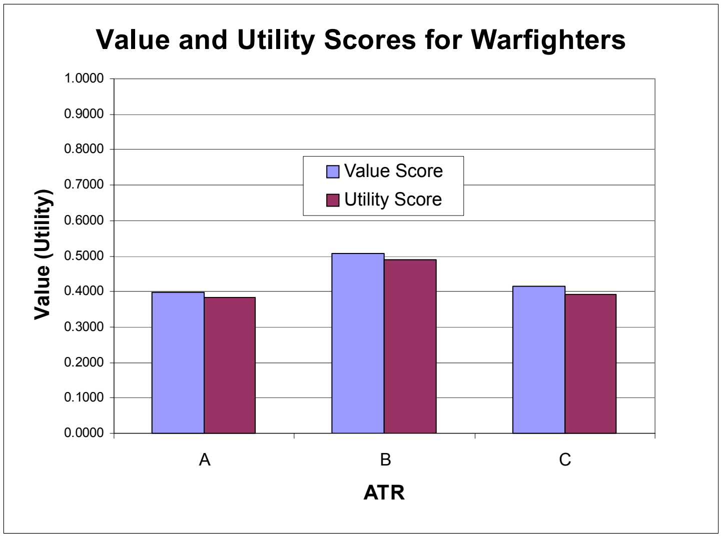


Figure 4.9 Warfighter Value and Utility Scores.

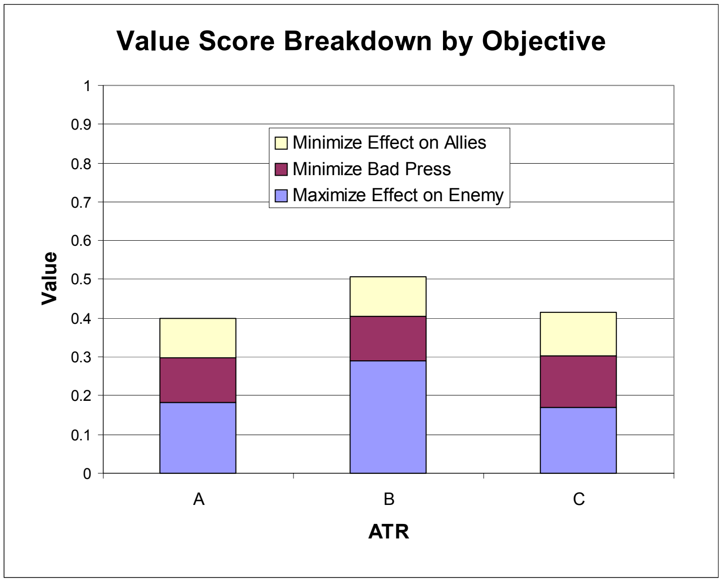


Figure 4.10 Warfighter Value Score Breakdown by Objective.

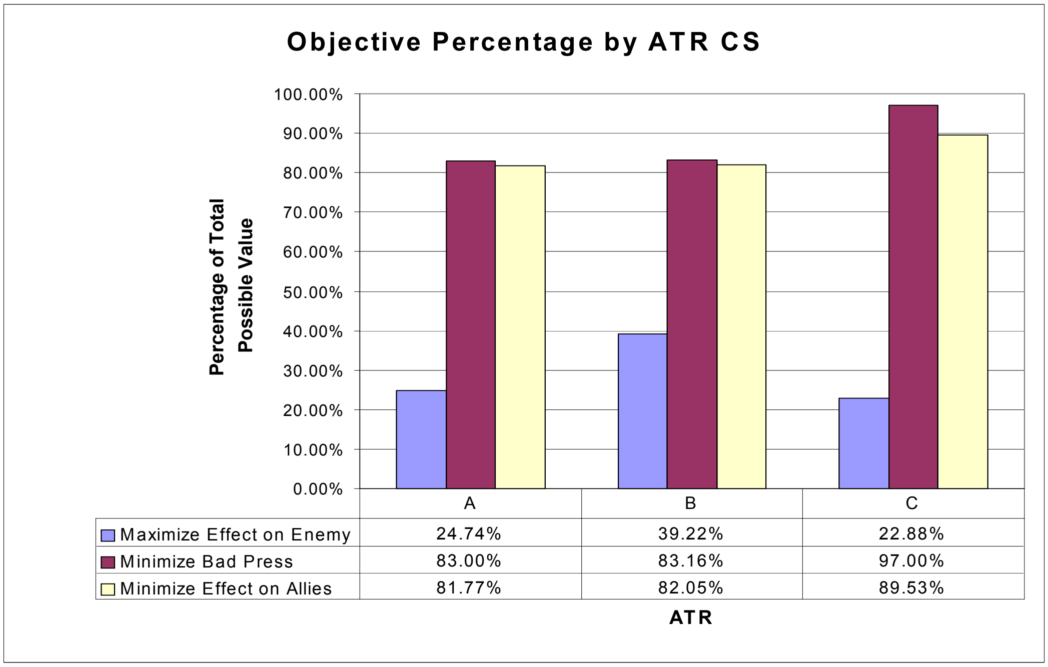


Figure 4.11 Warfighter Objective Percentage.

4.5 Analysis of Results

As mentioned in the previous chapter, the two DA frameworks were elicited at different times, and thus, constructed in different ways. The evaluator treats the ISR and CID employment profiles as two different branches of the DA model while the warfighter makes the two inseparable as they are combined in the combat model execution. Therefore, to remedy this situation, the evaluator ISR and CID results are given equal weighting and averaged to produce an overall value score, i.e. the *Total Value* seen in Table 4.3 and the graphs that follow. This assumption is reasonable and permits the direct comparison of the value scores between the evaluator and warfighter perspectives.

There are several different ways to analyze the results. Graphically, the values and utilities from the two frameworks can be compared side by side, as in Figures 4.12 and

4.13. The most important find is that both view the ranking of the three ATR CSs in terms of value and utility in a different way. The evaluator model indicates that ATR C is the best performer, while the warfighter model lists ATR B as the best. The graphs also illustrate the notion that the warfighter DA framework views the value scores of the ATR CSs less favorably than the evaluator framework, but is closer in agreement to the evaluator model in terms of utility. One argument for ATR A being the best overall ATR system is that it had the highest overall *AUC* measure average (Table A.2). However, neither the warfighter nor the evaluator framework selected ATR A as the best ATR. This indicates that the *AUC* measure, which is highly regarded in selecting the better system among the evaluation community, may not be the major factor in determining the superior system.

An important way to gain insight into the frameworks is to examine the results for differences. Table 4.6 indicates how the frameworks register similarities and differences between the ATRs and between the two frameworks. Notice that the largest discrepancy between the two frameworks occurs over the value and utility scores for ATRs C. When viewing the discrepancies between the two models, there are a few aspects to remember. First, the evaluator's model accounts for the system's employment concept, or ease of use. EADSIM, in this study, ignores the impact of an ATR employment impact. Also, ATR C performs better in terms of robustness when compared to the other systems. The evaluator's model places value on this trend, while the low resolution of EADSIM scenario used to evaluate the systems does not allow incorporation of this performance feature into the warfighter model.

As for similarities between the ATR CSs, it is difficult to find any agreement between the two frameworks. The warfighter model indicates that ATRs A and C are the closest in value, while the evaluator model indicates that ATRs A and B are the most similar. Between the two frameworks, ATR B is the nearest in terms of value and utility to both the warfighter and the evaluator.

Finally, cost-effectiveness is considered the “ultimate measure of goodness” when conducting an analysis of alternatives (24). Thus, the ATRs are plotted against their respective redeployment costs in Figures 4.14 and 4.15. Plotting an imaginary line between the two endpoints for a given framework (ATRs A and C, in this case) provides a benchmark for the ATR CSs in-between. Any point lying below this line does not provide a better value to cost ratio, and is, therefore, not cost-effective. It is clear that ATR B, in terms of both value and utility, is not cost effective to the evaluator, but is the most cost-effective to the warfighter.

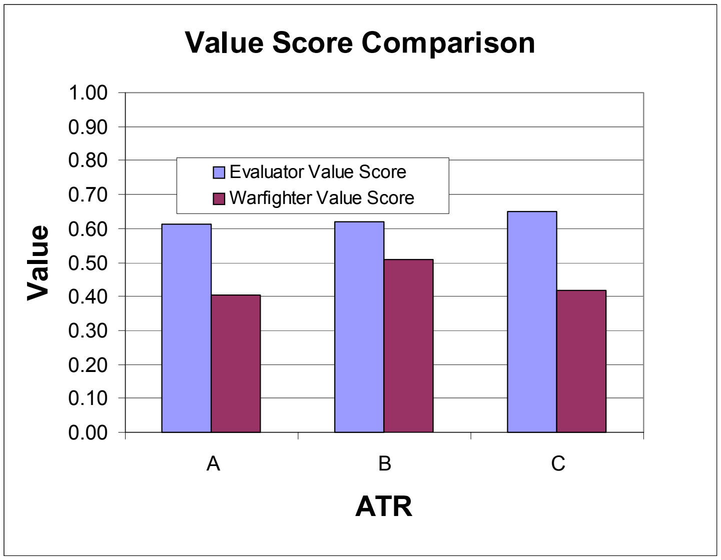


Figure 4.12 Graphical Comparison of Evaluator and Warfighter Value Scores.

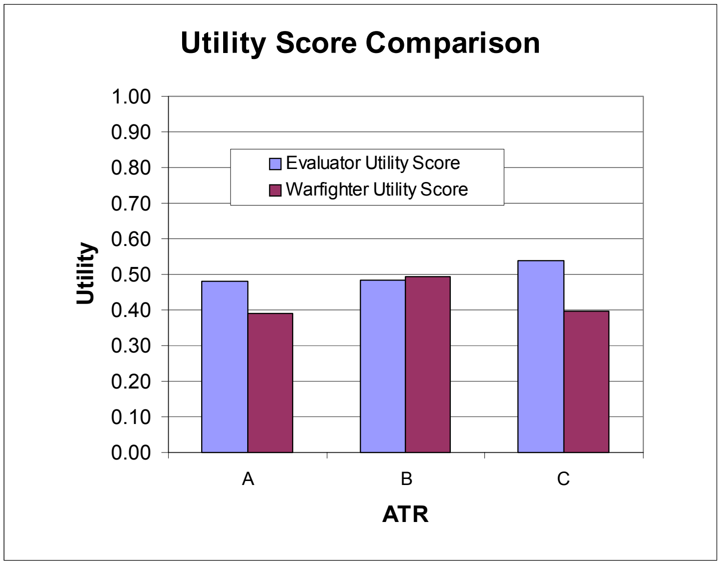


Figure 4.13 Graphical Comparison of Evaluator and Warfighter Utility Scores.

Table 4.6 Internal and External Differences of DA Frameworks.

Absolute Differences in Value Between ATRs By Model	A-B	A-C	B-C
Evaluator	0.0079	0.0367	0.0287
Warfighter	0.1023	0.0124	0.0899
Absolute Differences in Value Between Models By ATR	**A**	**B**	**C**
Evaluator-Warfighter	0.2063	0.1119	0.2306
Absolute Differences in Utility Between ATRs By Model	**A-B**	**A-C**	**B-C**
Evaluator	0.0021	0.0584	0.0564
Warfighter	0.1013	0.0050	0.0963
Absolute Differences in Utility Between Models By ATR	**A**	**B**	**C**
Evaluator-Warfighter	0.0900	0.0093	0.1434

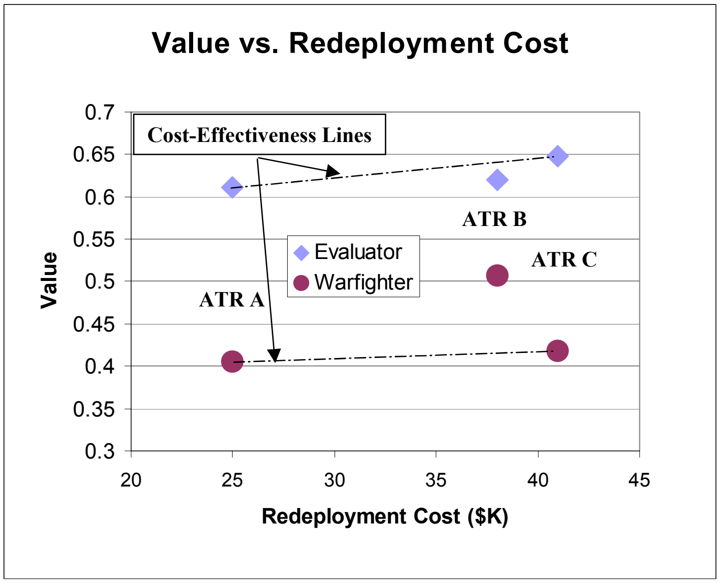


Figure 4.14 ATR Value Versus Redeployment Cost.

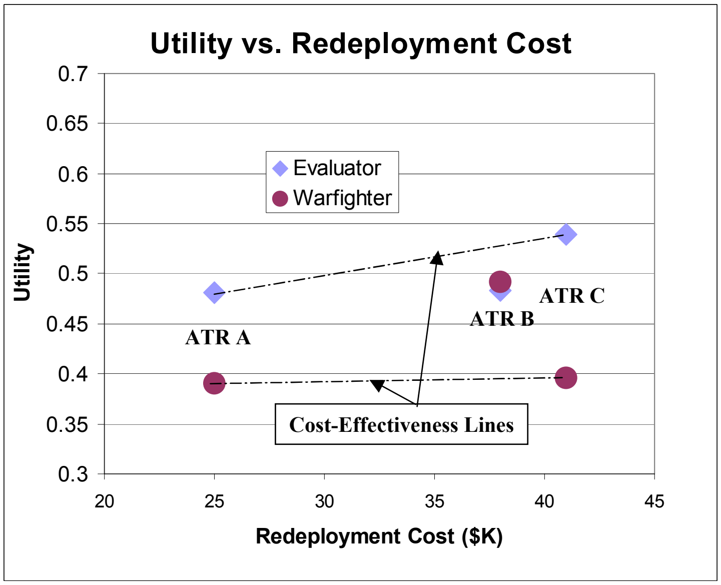


Figure 4.15 ATR Utility Versus Redeployment Cost.

Thus, given the results, a decision-maker could make a decision using the insights provided by the two different DA frameworks. The results at the very least indicate that the two parties are in disagreement over the ranking of the value and utility of the ATR CSs. The warfighter framework indicates that ATR C should be the first ATR CS to release from further consideration and that ATR A should be next, unless minimizing cost is the main objective of the ATR technology development process. The evaluator, on the other hand, indicates that ATR B is the prime candidate for exclusion from further comparison as it is the least cost-effective solution and that ATR C is the best performer. As for ATR C, the results from the warfighter model indicate that it is the worst system for impacting the enemy in the given scenario, but is the best for protecting friendly and neutral targets (Figure 4.11). Finally, the decision-maker may refuse to make a decision because the values of the system fail to meet a given threshold, or because he wishes to create a higher resolution scenario for more accurate comparison. Regardless of the decision, the decision-maker now has insight into what each party values in an ATR system and which MOEs and MOPs drive those preferences. The decision-maker also has a tool, based upon evaluator and user preferences, costs, risks, and performance characteristics, with which to compare various ATR systems and to justify decisions made throughout the ATR technology lifecycle.

V. Sensitivity Analysis

5.1 Overview

The previous chapter details the steps for calculating a single value/utility score from both the evaluator and the warfighter perspective, which may then be compared. Essentially, we have created a two-pronged DA model. As seen in Figure 2.27 and reproduced here in Figure 5.1, standard practice dictates a need for sensitivity analysis. Further, the analysis presented thus far is an expected value analysis in that expected values of the MOPs are propagated throughout the two prongs (Figure 5.2) while a value/utility score is realized (This value/utility score is *not* the expected value of the value/utility. This difficulty will be addressed in Chapter VI). With this in mind, sensitivity analysis could be performed using traditional DA tornado diagrams; alternatively, a partial differentiation approach is suggested below. Partial differentiation allows the calculation of the individual impact that each MOP has upon the value score at a given MOP setting. Thus, not only could an analyst use the MOP partial differentiation results to see *how much* an MOP is contributing to the value score via the magnitude of the partial differentiations, but also *in which direction* via the sign of the partial differentiations.

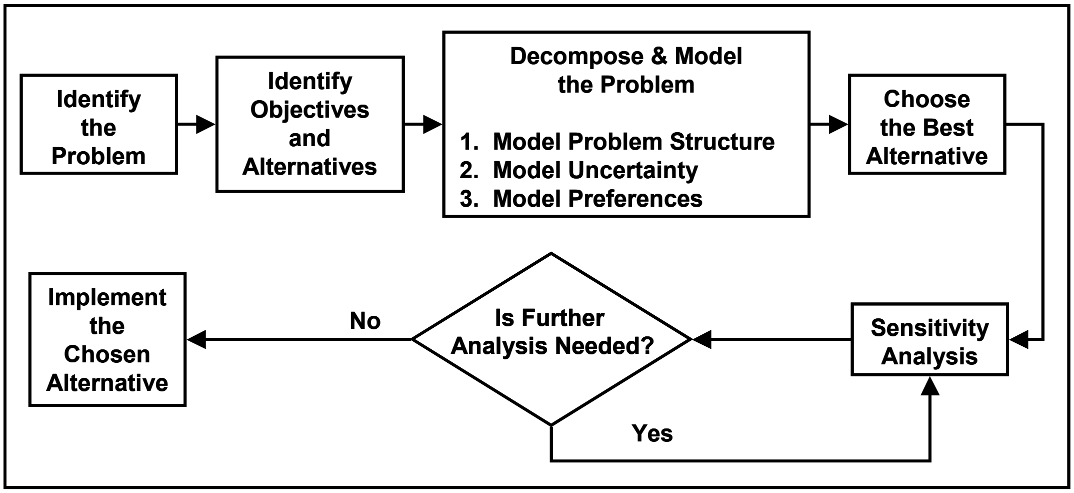

Identify the Problem
Identify Objectives and Alternatives
Decompose & Model the Problem
1. Model Problem Structure
2. Model Uncertainty
3. Model Preferences
Choose the Best Alternative
Sensitivity Analysis
Is Further Analysis Needed?
Yes
No
Implement the Chosen Alternative

Figure 5.1 Decision Analysis Process Flowchart (19).

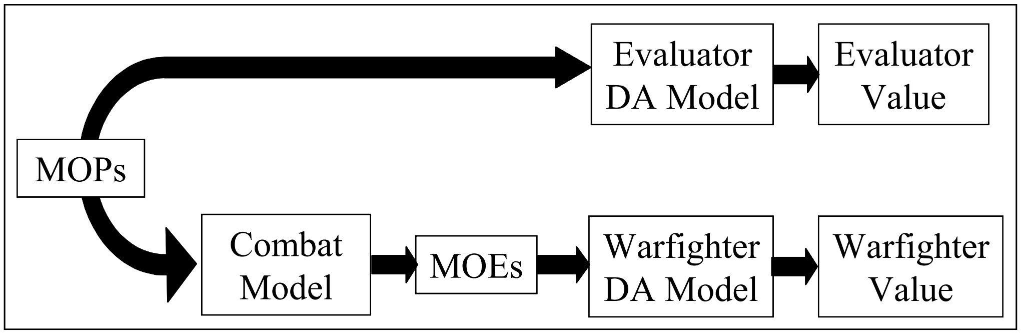


Figure 5.2 Two-Pronged DA Model Approach.

The problem with direct partial differentiation lies in the fact that the combat model, EADSIM, is not differentiable. There is no way to mathematically represent the internal working of this large combat model directly. However, using linear regression, a response surface may be generated that maps the input space (in this case, a given collection of MOP inputs) to a response surface (the resultant value scores). This response surface is differentiable. To facilitate the creation of this response surface, a surrogate combat model that emulates EADSIM performance is designed. This approach was adopted for several reasons. First, the scenario and data used within EADSIM is notional. Thus, the given scenarios, being of an unclassified nature, are of little operational interest. Therefore, there is no overwhelming need to use EADSIM for this analysis. Secondly, EADSIM is a very detailed combat model that offers several different controls to many of the aspects of the wargame. The surrogate, on the other hand, provides direct control over only the actions of interest within the combat scenario. Finally, the surrogate model requires much less computational time than EADSIM.

This chapter is organized as follows. After sensitivity analysis in the traditional decision analysis sense is performed, i.e. expected value tornado diagrams, a surrogate combat model is described. Next, a design of experiments is constructed which allows a mapping of MOPs to the MOEs, values, and utilities using the surrogate combat model results. This process produces a meta-model, i.e. a model of the surrogate model. With the meta-model, an ATR evaluator may easily generate expected value/utility rankings from a warfighters' perspective without the use of a combat model and use the information when making a decision concerning ATR technology development. Once the meta-model is inserted in place of the surrogate model, partial differentiation is performed. This produces changes in value per unit change in a MOP at a particular instantiation of the MOP space, which enables sensitivity analysis.

5.2 Tornado Diagrams

Expected value tornado diagrams are typically used to illustrate potential changes to the overall value and decision policies as inputs to the DA model, in this case individual

MOPs for the evaluator and MOEs for the warfighter, are varied (19). For example, as an MOP is varied away from its original value, the changes in overall value are reflected graphically via the tornado diagram, which allows an analyst to evaluate how the decision policy changes, i.e. selecting another ATR CS as the best system, by determining the most important factors in a decision. A tornado diagram, as in Figure 5.3, is composed of bars that indicate the variation in value, represented along the *x*-axis, as the variable in question is adjusted. The feature that produces the most variation in expected value for the decision generates the longest bar and is always presented at the top of the diagram. A *tornado*-like graphic is produced as subsequent, ranked value variation bars are added below the first. Changes in color along a bar indicate where a decision policy change is warranted, e.g. when a selection is no longer optimal. The vertical line represents the original, or base case, value with no variation in the inputs.

Tornado diagrams for the evaluator model, shown in Figures 5.3 and 5.4, indicate that varying the value of the inputs along their range of possible values could affect the optimal decision policy, noted by the bars that extend to the left of the solid line. The solid line represents the value of the second-best ATR CS in terms of value. Thus, when a change in an input associated with the best ATR CS results in a value score that lies below the competing ATR CS's value score, a decision policy change would be warranted. The diagrams also provide insight into the important inputs to the DA model. These could suggest areas that an ATR evaluator should spend more time and money during ATR technology development. Notice that the inputs which impact the value score as they are varied correspond to the higher ranked MOPs in terms of total possible value attributable (Figure 3.1). Figure 5.5 offers the results for the warfighter's perspective. In this case, the only possible policy decision change would occur if the percentage of Enemy C2 systems were to decrease to an unsatisfactory level. In other words, ATR B remains the best alternative in regards to value unless the percentage of enemy C2 systems damaged falls too low. Notice that this input is the highest ranked MOE in terms of total possible value (Table 3.5). Further insight may be gained by realizing that the top two inputs involve the *Maximizing Effect on Enemy* objective, which is the most heavily valued objective (Figure 3.12). Notice that the tornado diagrams for the evaluator and warfighter perspectives may not be compared directly as the evaluator analyzes the MOP set of inputs while the warfighter examines the set of MOE inputs. A direct comparison of tornado diagrams could be accomplished, in concept, by varying the individual MOPs, one at a time, to the minimum and maximum of their domain values, introducing the settings to a combat model, recording the change in value at each setting, and depicting the ranges in value for each MOP via a tornado diagram. This technique, however, is not accomplished here.

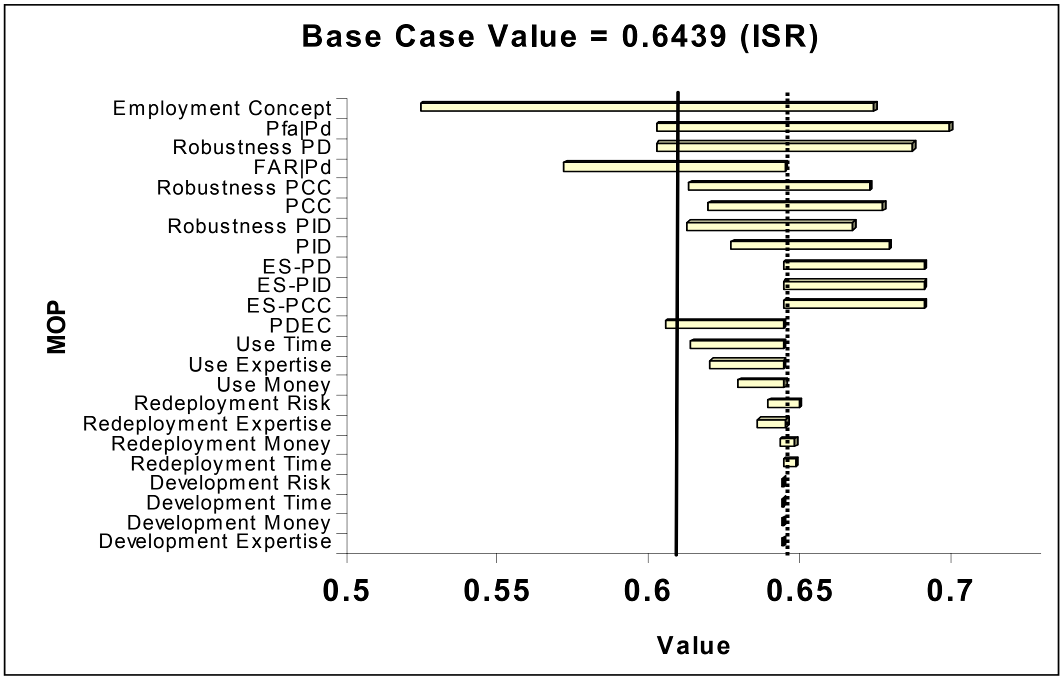


Figure 5.3 Evaluator Tornado Diagram (ISR).

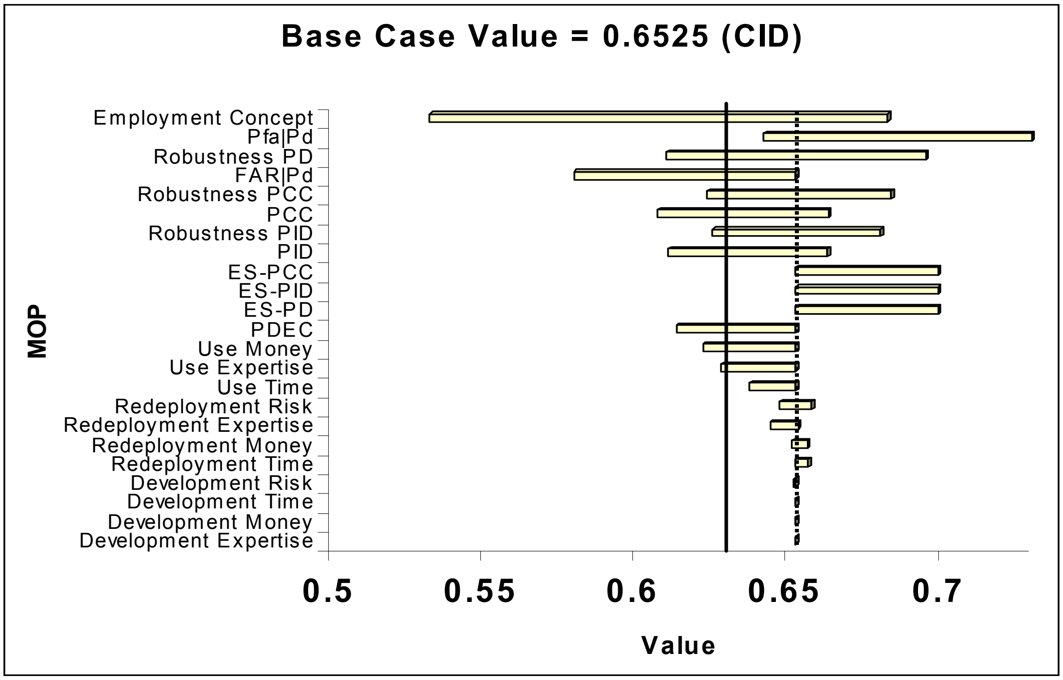


Figure 5.4 Evaluator Tornado Diagram (CID).

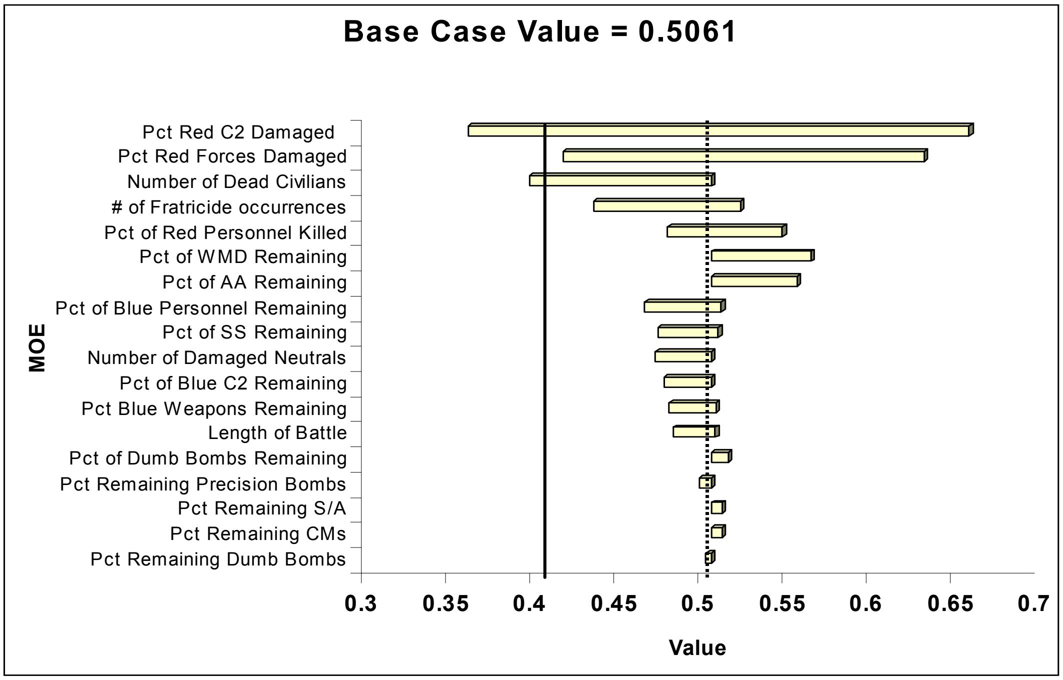


Figure 5.5 Warfighter Tornado Diagram.

The major weakness of the tornado diagram approach is that inputs can only be varied one at a time. Therefore, changes in input combinations and their effect on value cannot be realized via this method. For instance, features that may be inherently linked with one another, such as the *Percentage of Red Forces Damaged* and *Percentage of Dumb Bombs Remaining*, may be varied one at a time to realize changes in value, but there exists a cause and effect relationship between the two. The relationship is that inflicting casualties and damage requires weapons. Also, when using this method, it is important to vary inputs through realistic values. For instance, it is unrealistic to vary the *Length of Battle* feature to zero days because, under the assumption that a battle is going to take place, the battle will take time. Thus, while useful, the tornado diagram is not without limitations.

5.3 Combat Model Surrogate

A combat model, which closely emulates the performance processes of EADSIM, referred to here as the surrogate model, is used for this area of research. The surrogate combat model is in the form of nine Matlab® subroutines (produced in Appendix B): EADSIM2, Scenario, ISR, ATO, CID, BDA, Map, Map2, and Stats. Figure 5.6 depicts the subroutine sequence in the execution of the surrogate model. The EADSIM2 subroutine takes six arguments and returns combat results. The user inputs the number of air-to-ground forces as well as the number of ground objects found within the battle scene: Red_Tanks, MRLS, Intel, Blue_Tanks, and Neutral. The user also specifies the MOPs associated with the ATR being modeled within the scenario. The confusion matrices in Tables 5.1, 5.2, and 5.3 illustrate the way in which ATR MOPs are instantiated within the surrogate model. Table 5.1 indicates that each object within the battle scene has a 90% chance of being detected when the ATR operates under the ISR employment setting. The matrices in Tables 5.2 and 5.3 provide a probability that an object within the image scene, i.e. an MRLS, will be classified as a particular type, i.e. as an MRLS or a Red_Tank, for the respective employment setting. For example, if an MRLS system has been detected and the NCTR random number draw is 0.79534, the ISR system would declare the MRLS as an Intel Truck, while the CID system would declare that the target is indeed an MRLS. Each row is equivalent to the probability of identification matrix used in EADSIM (Table 4.2) except that the probabilities are expressed cumulatively.

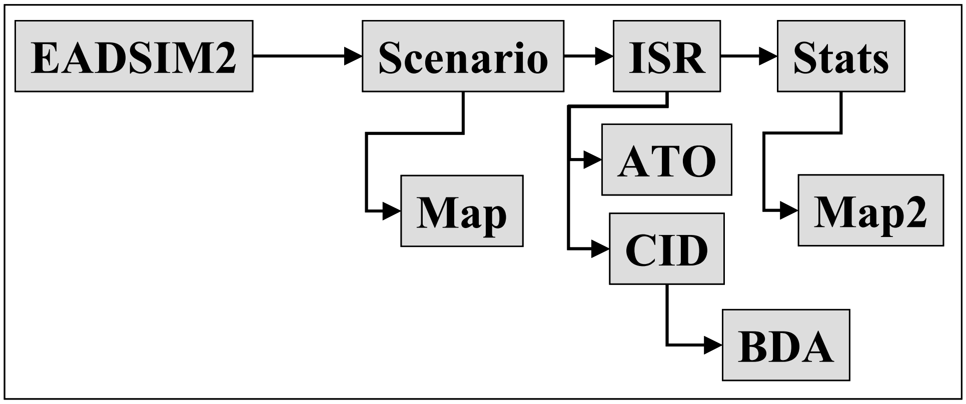


Figure 5.6 Surrogate Model Subroutine Sequence.

Table 5.1 ATR CS P_D Matrix in Surrogate Model.

Employment Setting	Red_Tank	MRLS	Intel	Blue_Tank	Neutral
ISR	0.9	0.9	0.9	0.9	0.9
CID	0.6	0.6	0.6	0.6	0.6

Table 5.2 ATR CS P_{ID} Matrix in Surrogate Model (ISR, $P_{ID\text{-}Type}$ = 0.70).

		Classified As				
		Red_Tank	**MRLS**	**Intel**	**Blue_Tank**	**Neutral**
Actual Type	**Red_Tank**	0.70	0.72	0.75	0.95	1.0
	MRLS	0.08	0.78	0.87	0.97	1.0
	Intel	0.03	0.13	0.83	0.88	1.0
	Blue_Tank	0.15	0.17	0.20	0.90	1.0
	Neutral	0.03	0.12	0.22	0.30	1.0

Table 5.3 ATR CS P_{ID} Matrix in Surrogate Model (CID, $P_{ID\text{-}Type}$ = 0.95).

		Classified As				
		Red_Tank	**MRLS**	**Intel**	**Blue_Tank**	**Neutral**
Actual Type	**Red_Tank**	0.95	0.96	0.96	0.99	1.0
	MRLS	0.02	0.97	0.98	0.99	1.0
	Intel	0.01	0.03	0.98	0.98	1.0
	Blue_Tank	0.04	0.04	0.04	0.99	1.0
	Neutral	0.00	0.02	0.04	0.05	1.0

The EADSIM2 subroutine begins the model execution by initializing the main variables and starting the main execution loop. The Scenario module randomly places ground objects in the battle scene and calls of the Map subroutine to display them in the first figure. Next, the ISR module emulates the processes of the ISR platform within the EADSIM model. The ISR platform attempts to detect and classify the ground targets within its range according to the matrices that capture its performance level. The ATO module then acts as the ground commander as it translates the targeting information generated by the ISR platform into an air tasking order, or strike list. The strike list is then passed to the fighters, which implement a sensor operating under a CID mode. The fighters must classify the potential target at the same identification level as the ISR platform in order to launch a weapon against that target. A target may be destroyed or damaged by the air-to-ground fighter. The Map2 module plots the effects of the air-to-ground attacks. The Stats module captures the numerical results of the combat model and translates them into MOEs. The BDA module simply updates a target status matrix.

5.4 Linear Regression

5.4.1 Design of Experiments.

Several issues must be answered before constructing the design of experiments for mapping the MOPs to both the warfighter's value scores and MOEs. First, the amount of runs must be manageable. Next, the confidence intervals surrounding the resultant value score estimates should be small enough to realize significant differences between sample runs. In other words, one surrogate model run at a given MOP setting will not offer the required amount of confidence to make judgments against another run. An acceptable confidence interval length must be generated.

The large number of MOPs forced the decision to choose a design of experiments that used a small fraction of design points. Rather than evaluating the surrogate model at 11,534,336 different samples (2^{20}), a fractional factorial design allowed the use of 32 different runs to evaluate the various factors impacting the MOEs and values. The design consisted of 20 separate factors with 15 different design generators (or confounding rules) where no main effect is aliased with any other main effect, but is aliased with at least a two-factor interaction; that is, a 2^{20-15} fractional factorial design of resolution III (48). The fractional factorial design is listed in Table A.6.

To address the sample size issue, random draws were taken from a set of MOP estimates, with a standard deviation of 0.05 for each MOP and the following means: P_{D-ISR} = 0.9, P_{D-CID} = 0.6, P_{ID-ISR}= 0.7, and P_{ID-CID} = 0.95. Using the same scenario introduced in Chapter IV (15 Red_Tanks, 7 MRLSs, 4 Intel Trucks, 12 Blue_Tanks, 15 Neutral vehicles, and 5 Blue AG_Attackers), the MOP samples were introduced to the surrogate model in run increments of 20 from 20 samples to 400 samples to calculate the amount of variance evident between run increments. This produced 40 different observations. Thus, the value score of the first observation is the average of 20 value scores, the second observation is a value score based on the average of 40 value scores, and so forth. Confidence intervals, based upon a confidence level of 95%, surrounding the mean value for each of the observations were calculated. Table 5.4 lists a portion of the results.

Table 5.4 Surrogate Model Incremental Run Results.

Observation	Runs	Value Score Mean	Value Score Variance	95% CI Half-Length
1	**20**	0.7151	0.0113	0.0466
2	**40**	0.696	0.0105	0.0318
3	**60**	0.6951	0.0151	0.0311
4	**80**	0.6835	0.0107	0.0227
5	**100**	0.6753	0.0147	0.0238
6	**120**	0.6858	0.0120	0.0196
7	**140**	0.6864	0.0120	0.0181
8	**160**	0.6842	0.0120	0.0170
9	**180**	0.6735	0.0136	0.0170
…	**…**	…	…	…
20	**400**	0.6864	0.0122	0.0108

Next, the value score differences between each of the 20 observations were calculated, resulting in 190 different value score differences. The confidence interval half-lengths of the 20 run increments were then compared against the value score distance lengths. To discern between 90% of the value score difference would require selecting a confidence interval half-length of 0.00125 (0.0025 divided by 2), which corresponds to the first bin and its associated frequency in Figure 5.7. Figure 5.8 illustrates the confidence interval lengths surrounding each observation as the run number increase. Thus, according to the results, to achieve an acceptable confidence interval half-length would require a large number of runs.

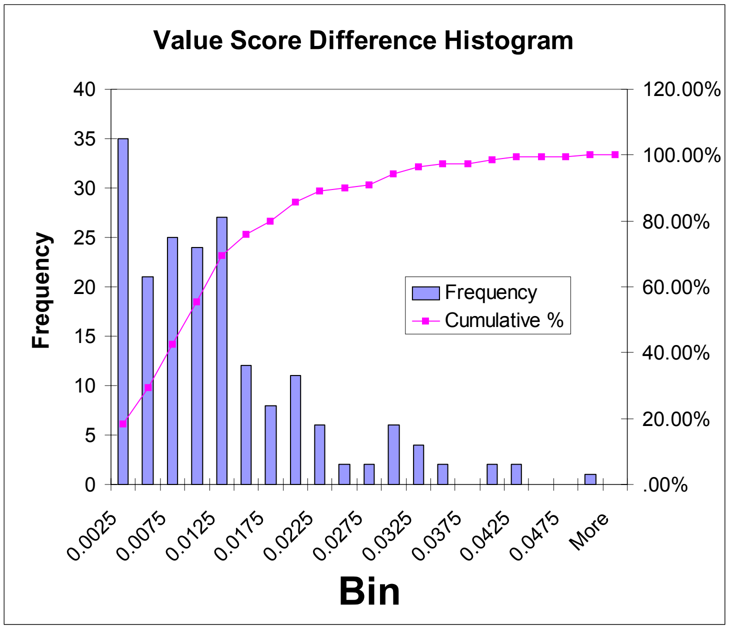


Figure 5.7 Value Score Difference Histogram.

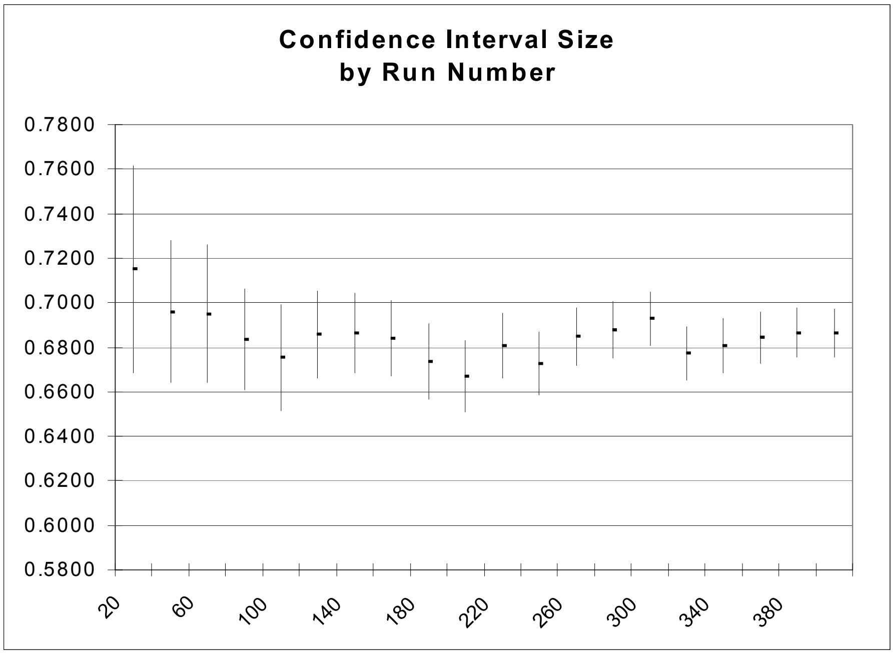


Figure 5.8 Confidence Interval Size by Run Number.

To solve for the required sample size needed to achieve a given value score distance, the following heuristic could be implemented (10). Given *n* pilot runs (samples) for *j* value score realizations (systems), the half width of the corresponding confidence interval could be computed as:

$$HW = s_{pooled} \cdot \sqrt{\frac{1}{n_1} + \frac{1}{n_2}} \cdot t_{n_1+n_2-1,(1-\frac{\alpha}{2})}, \tag{5.1}$$

where

$$s_{pooled} = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}} \tag{5.2}$$

is the pooled standard error of the specified point estimator, n_1 and n_2 correspond to the sample sizes of the *j*th observations, and $t_{n1+n2-1,(1-\alpha/2)}$ is the 100(1-α/2) percentage point of a *t* distribution with n_1+n_2-1 degrees of freedom. Assuming that $n = n_1 = n_2$, then the equation reduces to:

$$HW = s_{pooled} \cdot \sqrt{\frac{2}{n}} \cdot t_{2n-1,\left(1-\frac{\alpha}{2}\right)}. \tag{5.3}$$

Then, the upper and lower values of a confidence interval surrounding the value score difference could be computed as:

$$L_U = (\bar{x}_1 - \bar{x}_2) + s_{pooled} \cdot \sqrt{\frac{2}{n}} \cdot t_{2n-1,\left(1-\frac{\alpha}{2}\right)}, \tag{5.4}$$

and

$$L_L = (\bar{x}_1 - \bar{x}_2) - s_{pooled} \cdot \sqrt{\frac{2}{n}} \cdot t_{2n-1,\left(1-\frac{\alpha}{2}\right)}, \tag{5.5}$$

where the first term in the equation represents the difference between the respective value score means for two different systems. If $L_L \leq 0 \leq L_U$ is false, then there is a significant difference between the value score means of the two systems. If $L_L \leq 0 \leq L_U$ is true, then we define:

$$D = Abs(\bar{x}_1 - \bar{x}_2), \tag{5.6}$$

or the absolute difference between the value score differences. Of the various differences, we may now define:

$$N_{\alpha}^{*}(D) = \min\{i \geq n: \quad s_{pooled} \cdot \sqrt{\frac{2}{i}} \cdot t_{2n-1,\left(1-\frac{\alpha}{2}\right)} \leq D \ \} \tag{5.7}$$

where i is the sample size needed to make the expression an equality. Solving this equation provides the sample size necessary to distinguish between all value score differences between the j realizations. These equations work if the sample variances are assumed to be equal and the sample sizes are the same, i.e. $n_1 = n_2$. As an example using the surrogate model, with a pooled variance of 0.0108 and a t-statistic with $\alpha = 0.05$ and 99 degrees of freedom (where $n_1 = n_2 = n = 50$), 522 runs would be required to distinguish a significant difference between a value score mean difference of 0.0132.

However, it may be more beneficial to solve for *practically* significant confidence intervals surrounding the value score difference means rather than *statistically* significant intervals (10). In other words, the computational time required to derive a difference between to value score differences may not be worth the end result. Therefore, a predetermined D, from equation 5.6, should be selected in order to find the proper sample size. Conversely, an acceptable sample size should be selected, and the value score difference detectable may be ascertained from Equation 5.7. This method is implemented for the remainder of this study as $n = 300$ surrogate model runs were used for all appropriate analyses, unless otherwise noted.

5.4.2 Model Building.

A linear regression model can now be used to capture the relationship between the MOPs and both the MOEs and the value/utility scores. First, a model using only the first order terms without interactions is used to find important features (MOPs) that best account for the variation found within the outputs (MOEs or value). The best features (effects) are retained and goodness-of-fit tests are performed. If the model is unsatisfactory, linear regression is performed again using interaction terms. Again, the best features are retained and goodness-of-fit tests are performed. If the interaction model is unsatisfactory, then second-order terms are included within the model. The best model of the three models is used to capture the MOP-MOE and MOP-value transformation for each output.

The following methodology was used to select the effects included in the regression models. First, a backward stepwise model building approach is applied. With each MOP included in the model, effects are systematically dropped out to minimize the value of the Akaike's Information Criterion (*AIC*) defined as:

$$AIC = n \cdot \ln(SSE / n) + 2p \,, \tag{5.1}$$

where n is the number of observations, p is the number of model parameters including the intercept, and *SSE* is the sum of squared errors. This is a general criterion for choosing the best number of parameters to include in a model. The model that has the smallest value of *AIC* is considered the best. Next, beginning with the *AIC*-minimum model, effects that have a *t-ratio* in absolute value less than 2.0 are retained to create the *t-ratio* model. Thus, with the *t-ratio* model, we can be certain (with a confidence level of 95%) that the weight given to an effect in the model is significantly different from zero (61).

Two additional techniques were available should the previous methods provide unsatisfactory results. The *Cp* model uses Mallow's *Cp* as a model selection criterion by selecting the model where *Cp* approaches the number of parameters in the model, *p*. Mallow's *Cp* criterion is an alternative measure of total squared error defined as:

$$Cp = \left(\frac{SSEp}{s^2} \right) - (N - 2p) \tag{5.2}$$

where s^2 is the MSE for the full model, *SSEp* is the sum-of-squares error for a model with *p* variables and the intercept, and *N* is the number of observations (61). Finally, the *adjusted* R^2 value was minimized using the backward stepwise model building approach. R^2 is the proportion of the variation in the response that can be attributed to terms in the model rather than to random error, i.e.,

$$R^2 = \frac{SSR}{S_{YY}} = 1 - \frac{SSE}{S_{YY}} . \tag{5.3}$$

The *adjusted* R^2 term, written as *Adj-*R^2 or R^2_{adj}, adjusts the R^2 term to make it more comparable over models with different numbers of parameters by using the degrees of freedom in its computation.

$$R^2_{Adj} = 1 - \left(\frac{n-1}{n-p} \right) (1 - R^2) \tag{5.4}$$

Unlike R^2, which always increases as more terms are included in the model, R^2_{adj} is useful in stepwise model-building procedures as it decreases when unnecessary terms are added to the model (61). Figure 5.9 uses an abstract illustration to describe the concept behind mapping the MOPs directly to the MOEs generated via a combat model. The arrow refers to the transformation process to be modeled via mathematical techniques. Thus, *T*, which is a transformation matrix produced via the linear regression model building process, is a means to estimate the effects of the combat model. The results of the model building methodology are given below in Table 5.5. The model type column refers to the technique used to generate the data within the given row. Rows in bold typeface indicate the model selected for use.

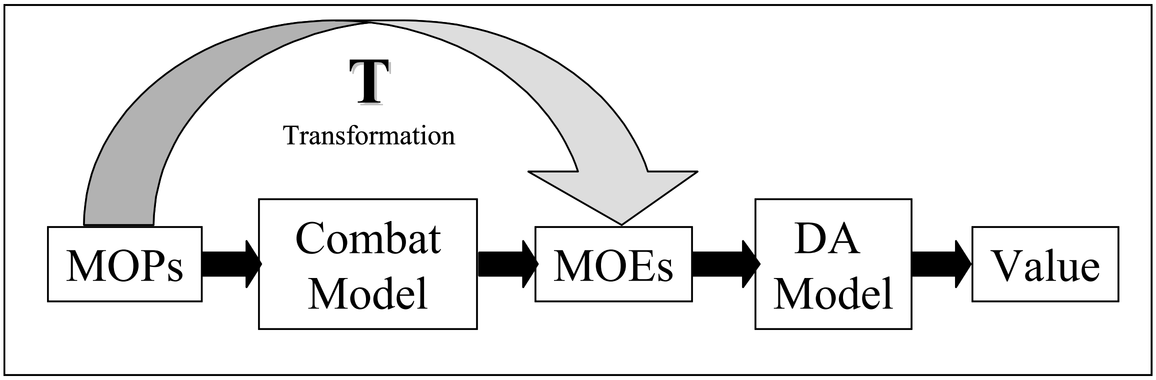


Figure 5.9 Abstract Depiction of the MOP to MOE Mapping Concept (Warfighter).

Table 5.5 MOPs to MOEs Linear Regression Results.

MOP	Model Type	# of Features	Whole Model F-Ratio	Adj R^2	RMSE	AIC
Pct of S/S Missiles Remaining	***AIC***	**9**	**62.12**	**0.9466**	**0.00996**	**-286**
	RAdj	**9**	**62.12**	**0.9466**	**0.00996**	**-286**
	t-Ratio	6	72.06	0.9322	0.01123	-281
Pct Red Forces Damaged	*AIC*	16	210.77	0.9908	0.00522	-327
	RAdj	16	210.77	0.9908	0.00522	-327
	t-Ratio	**12**	**200.94**	**0.9872**	**0.00616**	**-316**
Pct of Red Personnel Killed	*AIC*	16	210.60	0.9908	0.00453	-336
	RAdj	16	210.60	0.9908	0.00453	-336
	t-Ratio	**12**	**200.38**	**0.9872**	**0.00535**	**-325**
Pct Red C2 Damaged	***AIC***	17	62.81	0.9713	0.01087	-280
	RAdj	17	62.81	0.9713	0.01087	-280
	t-Ratio	**12**	**64.88**	**0.9611**	**0.01266**	**-270**
Length of Battle	***AIC***	15	123.47	0.9834	0.19278	-96
	RAdj	15	123.47	0.9834	0.19278	-96
	t-Ratio	**13**	**124.29**	**0.9810**	**0.20615**	**-91**
Number of Dead Civilians	***AIC***	**11**	**9.12**	**0.7423**	**0.28008**	**-72**
	RAdj	**11**	**9.12**	**0.7423**	**0.28008**	**-72**
	t-Ratio	6	12.98	0.6987	0.30282	-70
Number of Damaged Neutrals	*AIC*	14	7.85	0.7558	0.07446	-156
	RAdj	14	7.85	0.7558	0.07446	-156
	t-Ratio	**6**	**12.21**	**0.6844**	**0.08465**	**-152**
# of Fratricide occurrences	*AIC*	12	14.98	0.8440	0.05044	-182
	RAdj	13	14.01	0.8451	0.05026	-182
	t-Ratio	**9**	**17.27**	**0.8252**	**0.05339**	**-180**
Pct Blue Weapons Remaining	*AIC*	**8**	**14.07**	**0.7714**	**0.00426**	**-342**
	RAdj	10	11.77	0.7765	0.00421	-342
	t-Ratio	5	22.86	0.7383	0.00456	-340
Pct of Blue Personnel Left	***AIC***	**7**	**30.96**	**0.8712**	**0.00193**	**-393**
	RAdj	**7**	**30.96**	**0.8712**	**0.00193**	**-393**
	t-Ratio	4	47.21	0.8564	0.00203	-392
Pct Remaining Dumb Bombs	*AIC*	16	107.79	0.9822	0.00042	-488
	RAdj	16	107.79	0.9822	0.00042	-488
	t-Ratio	**14**	**117.76**	**0.9814**	**0.00043**	**-486**
Pct Remaining Precision Bombs	*AIC*	14	89.24	0.9755	0.00049	-477
	RAdj	14	89.24	0.9755	0.00049	-477
	t-Ratio	**12**	**92.54**	**0.9726**	**0.00052**	**-474**

Using the linear regression results listed in Table 5.5, the MOPs associated with a given ATR CS may be used to produce estimates of the surrogate model MOEs. This is accomplished via matrix multiplication:

$$MOE = MOP \cdot T, \tag{5.5}$$

where *MOE* is the 1×18 vector of estimated MOE values, *MOP* is the 1×21 vector of ATR CS performance measures (including an intercept term), and *T* is the 21×18 transformation matrix composed of columns which contain the linear regression parameters associated with the selected models. These estimates may then be introduced to the value functions and weights of the warfighter's DA model. Thus, the above value score equation could be rewritten as:

$$Value_W(MOP) = \sum_{k=1}^{K} w_{kv}^{3}\left(\sum_{j=1}^{J} w_{jk}^{2}\left(\sum_{m=1}^{M} w_{mj}^{1}\left(g_m\left(MOP \cdot T\right)\right)\right)\right) \tag{5.6}$$

where $Value_W$ is the value score estimate for the warfighter, *MOP* is a given set of *n* MOPs, the w_{ij} are the *i*th weights associated with the *j*th branch in the DA framework, g_n is the *n*th value functions associated with each MOE and expressed as a polynomial up to the third order. The evaluator's DA framework may also be regressed, but it uses a simpler equation:

$$Value_E(MOP) = \sum_{k=1}^{K} w_{kv}^{3}\left(\sum_{j=1}^{J} w_{jk}^{2}\left(\sum_{m=1}^{M} w_{mj}^{1}\left(g_m\left(MOP\right)\right)\right)\right), \tag{5.7}$$

as there is no transformation from MOPs to MOEs. The value score estimate for the evaluator is $Value_E$. The g_m function for the evaluator uses a polynomial expression up to the second order.

Continuing in the same model building methodology, the MOPs may be used to directly estimate the value score for a particular ATR at a given setting and at a given scenario. Figure 5.10 uses an abstract illustration to describe the concept behind mapping the MOPs directly to the value score. This transformation could be represented by:

$$Value = MOP \cdot V, \tag{5.8}$$

where *V* is the transformation vector produced by linear regression of the MOP inputs (the *MOP* matrix) versus the surrogate model value scores, and *Value* is the estimated value score. A *Utility* estimate is produced similarly, using a different transformation vector, *U*. Table 5.6 details the results of applying the aforementioned linear regression techniques toward creating a model that accurately estimates the value score given the set of MOPs. Table 5.7 accomplishes the same for estimating the utility score with the same set of MOPs.

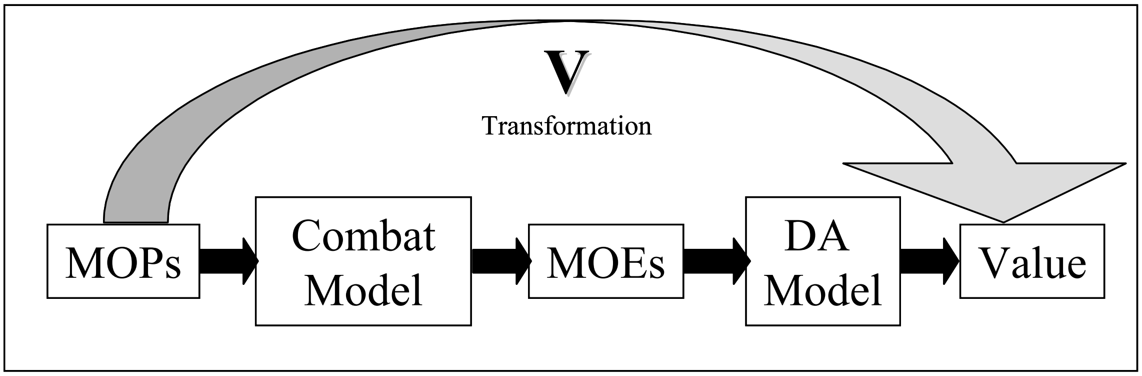


Figure 5.10 Abstract Depiction of the MOP to Value Mapping Concept (Warfighter).

Table 5.6 MOPs to Value Score Linear Regression Results (Warfighter).

Model Type	Number of Features	Whole Model F-Ratio	Adj R^2	RMSE	AIC
AIC	15	80.27	0.9746	0.00599	-318
RAdj	15	80.27	0.9746	0.00599	-318
t-Ratio	**11**	**93.57**	**0.9705**	**0.00646**	**-314**

Table 5.7 MOPs to Utility Score Linear Regression Results (Warfighter).

Model Type	Number of Features	Whole Model F-Ratio	Adj R^2	RMSE	AIC
AIC	14	104.84	0.9791	0.00564	-322
RAdj	14	104.84	0.9791	0.00564	-322
t-Ratio	**11**	**105.98**	**0.9739**	**0.00632**	**-315**

For the evaluator, this transformation process does not include a transformation through a combat model. This allowed a larger fractional factorial design to be used. The design made use of 128 different observations with 23 factors sampled at two different levels. Figure 5.11 abstractly illustrates the transformation vector as it transforms the evaluator's MOP set directly into value. Table 5.8 details the results of the linear regression model building process for the evaluator's transformation vector. The *AIC*, *R^2-Adj*, and *t-Ratio* models yielded a model consisting of all 23 factors. Due to confounding effects within model, this is not a preferred solution. To combat the confounding effects, a model, called the *7 Objectives* model, that consists of the strongest factor from each of the seven objectives (except for the *Classification Ability* objective, from which both MOPs were used as factors as they represented the two most important factors) was used.

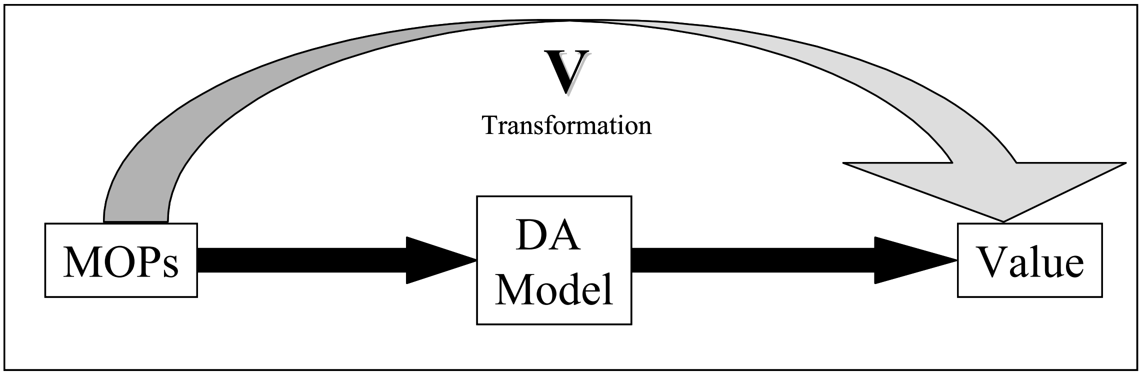


Figure 5.11 Abstract Depiction of the MOP to Value Mapping Concept (Evaluator).

Table 5.8 MOPs to Value Score Linear Regression Results (Evaluator).

Model Type	Number of Features	Whole Model F-Ratio	Adj R^2	RMSE	AIC
AIC	23	3.1601e+8	1.00	0.000017	-2786
RAdj	23	3.1601e+8	1.00	0.000017	-2786
t-Ratio	23	3.1601e+8	1.00	0.000017	-2786
7 Objs	8	92.4255	0.9521	0.050252	-757

5.4.3 Results.

The transformation matrix, T, provides a mapping of MOPs to MOEs, as depicted in Figure 5.9. Not only does this matrix serve as a method for transforming MOPs efficiently into MOEs without the need for a combat model, but it is also serves as a method for determining which MOPs are the most important for affecting MOEs. For instance, the transformation matrix, listed in its entirety in Table A.5, indicates that, for the *Pct Red Forces Damaged* MOE, the CID Red_Tank P_D (0.542) is the most significant factor. Thus, the most important factor in maximizing the effect on enemy forces is the ability of the fighter aircraft to locate the Red_Tanks in the battlespace. In fact, the second most important factor is the CID platform's ability to correctly identify Red_Tanks.

The direct transformation of MOPs to value and utility using a linear regression approach for the warfighter's perspective yields the transformation vectors found in Table A.4. As expected, the close relationship between the warfighter's expectation in value and utility is reflected in the transformation vector. Also, the magnitude for each of the MOPs is roughly the same for each vector.

The linear regression results can be used in the following manner. Table 5.9 lists the value score results for the evaluator's DA model: via direct MOP insertion into the evaluator DA model and via linear regression using the *All 23 Factors* and *7 Objectives* models. While the value magnitudes are different, the rankings are not. Thus, the evaluator could use the linear transformation from MOPs directly to value score as an estimate of the DA model. Table 5.10 lists the value score results for the warfighter's DA model: via introducing the MOP set into the warfighter's DA model employing both EADSIM and the surrogate model, using the linear transformation from MOPs to MOEs, and using the linear transformation from MOPs to value. The warfighter linear transformation from MOEs to value results in a different ranking scheme, but retains ATR B as the best performer.

The weakness in this approach seems to lie in the direct transformation from the MOPs to the value score. There are several possible reasons for the differences between the expected and the predicted value scores. First, the regression performed is linear, whereas the value and utility functions of the DA models are typically non-linear. Next, the surrogate model does not perfectly emulate the execution of EADSIM. Therefore, there will be some differences in the MOEs, which impact the transformation matrix and the resultant value scores produced through linear regression. Finally, the linear regression model building process does induce error into the predicted value score as it tries to fit the data. In other words, the predicted value score will always be different from the actual value score.

Table 5.9 Evaluator Linear Regression Comparison (Rank in Parenthesis).

ATR CS	**Evaluator DA Model**	**MOP to Evaluator Value (*All 23 Factors* Model)**	**MOP to Evaluator Value (*7 Objectives* Model)**
A	0.6116 (3)	0.5556 (3)	0.5632 (3)
B	0.6195 (2)	0.5668 (2)	0.5825 (2)
C	0.6482 (1)	0.5995 (1)	0.6042 (1)

Table 5.10 Warfighter Linear Regression Comparison (Rank in Parenthesis).

ATR CS	**Warfighter DA Model w/EADSIM**	**Warfighter DA Model w/Surrogate**	**MOP to MOE via LR (T)**	**MOP to Value via LR (V)**
A	0.3988 (3)	0.6372 (3)	0.6735 (2)	0.6889 (1)
B	0.5061 (1)	0.6514 (1)	0.6741 (1)	0.6708 (3)
C	0.4140 (2)	0.6428 (2)	0.6594 (3)	0.6813 (2)

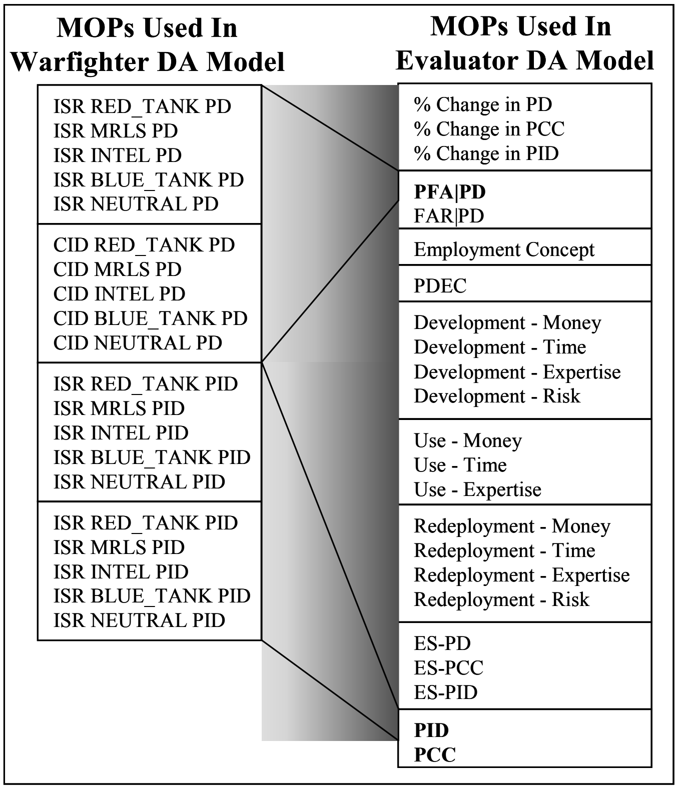


Figure 5.12 MOP Input Structure for Evaluator and Warfighter Frameworks.

5.5 Partial Differentiation.

One way to gauge the relative importance, or saliency, of an MOP to the resultant value or utility is partial differentiation. The value score from the warfighter's DA model may be computed from the formula:

$$Value(MOP_n) = \sum_{k=1}^{K} w_{kv}^{3} \left(\sum_{j=1}^{J} w_{jk}^{2} \left(\sum_{m=1}^{M} w_{mj}^{1} \left(g_m \left(\sum_{n=1}^{N} w_{nm}^{0} \cdot MOP_n \right) \right) \right) \right), \qquad (5.8)$$

where MOP_n is the nth of N MOPs, the w_{ij} are the ith weights associated with the jth branch in the DA framework, g_n is the nth value functions associated with each MOE and expressed as a polynomial up to the third order (19,39). Utility scores are produced similarly, using a utility function, h_n, rather than a value function, g_n. The w^0_{mn} weights were produced in the linear regression process (Formula 5.5) and take the form of the transformation matrix (Table A.5). The entire transformation is depicted graphically in Figure 5.13.

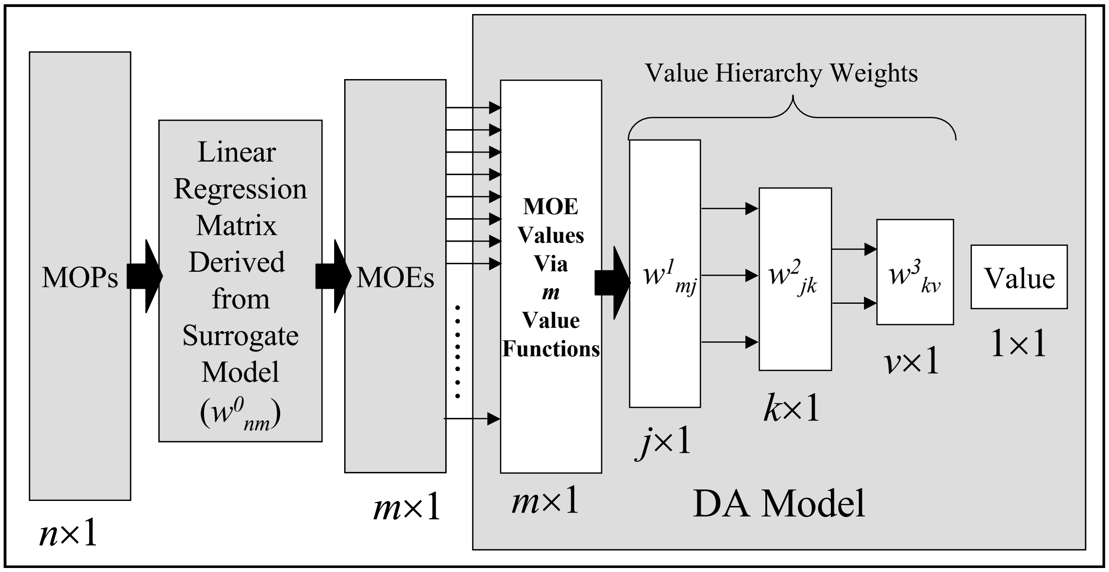


Figure 5.13 Graphical Depiction of Warfighter MOP to Value Transformation.

The value formula may be rewritten as:

$$Value(MOP_n) = \sum_{k=1}^{K}\sum_{j=1}^{J}\sum_{m=1}^{M} w^3_{kv} w^2_{jk} w^1_{mj}\left(g_m\left(\sum_{n=1}^{N} w^0_{nm}\cdot MOP_n\right)\right). \tag{5.9}$$

This equation may then be differentiated with respect to each individual MOP. Applying the chain rule here we have:

$$\frac{\partial Value}{\partial MOP_i} = \sum_{k=1}^{K}\sum_{j=1}^{J}\sum_{m=1}^{M} w^3_{kv} w^2_{jk} w^1_{mj}\left(\frac{\partial g_m}{\partial s_m}\cdot\frac{\partial s_m}{\partial MOP_i}\right), \tag{5.10}$$

$$\frac{\partial Value}{\partial MOP_i} = \sum_{k=1}^{K}\sum_{j=1}^{J}\sum_{m=1}^{M} w^3_{kv} w^2_{jk} w^1_{mj}\,\frac{\partial g_m}{\partial s_m}\cdot w^0_{im}\,, \tag{5.11}$$

where

$$g_m = b_{0m} + b_{1m} s_m + b_{2m} s_m^2 + b_{3m} s_m^3 \tag{5.12}$$

and

$$\frac{\partial g_m}{\partial s_m} = b_{1m} + 2b_{2m} s_m + 3b_{3m} s_m^2 . \tag{5.13}$$

The process is simpler for the evaluator MOP to value transformation since there is no transformation through a combat model. Thus, from the evaluator's perspective, Figure 5.13 would not include a transformation matrix to account for the combat model nor the MOEs.

The results for the warfighter, using the data from the application in Chapter IV, are given in Table 5.11. The sample MOP points are given, followed by the evaluated partial differentiations for the individual MOPs. The results indicate that the features that offer the most change in regards to increasing the value score are the P_D and P_{ID} performance measures for the CID platform operating against Red_Tanks, MRLSs, and Intel trucks. Increasing the CID platform's P_D performance against the MRLS vehicles offers the largest decrease in value, while the major increase in value is available through increasing the ISR platform's detection performance against friendly tanks. The former seems to be contradictory in that increasing detection and identification against the enemy results in a decreased value. However, this probably reflects the additional time required, the additional expended weapons, and the increased probability of destroying friendly and neutral targets as more sorties are performed. The latter seems to reflect the importance placed upon reducing fratricide, also seen in the warfighter tornado diagram (Figure 5.5). It is interesting that increasing the CID platform's P_{ID} performance against

Blue_Tanks (though small) results in a negative impact on the value score. This could be due to the fact that most Neutral vehicles are in a close proximity to Blue_Tanks, or that correctly detecting a Blue_Tank results in the slight increase in probability of an occurrence of fratricide due to the possible misclassification as an enemy vehicle.

Table 5.11 Warfighter Partial Differentiation Vectors at Sample MOP Observations.

	Observed MOPs			**Partial Differentiations**		
MOP	**A**	**B**	**C**	**A**	**B**	**C**
ISR RED TANK PD	0.900	0.900	0.900	-0.0104	-0.0104	-0.0103
ISR MRLS PD	0.900	0.900	0.900	0.0000	0.0000	0.0000
ISR INTEL PD	0.900	0.900	0.900	-0.0730	-0.0749	-0.0758
ISR BLUE TANK PD	0.900	0.900	0.900	**0.1073**	**0.1083**	**0.1098**
ISR NEUTRAL PD	0.900	0.900	0.900	-0.0002	-0.0002	-0.0002
CID RED TANK PD	0.600	0.600	0.600	**-0.1623**	**-0.1623**	**-0.1647**
CID MRLS PD	0.600	0.600	0.600	**-0.2793**	**-0.2787**	**-0.2832**
CID INTEL PD	0.600	0.600	0.600	**-0.2155**	**-0.2189**	**-0.2216**
CID BLUE TANK PD	0.600	0.600	0.600	0.0020	0.0020	0.0020
CID NEUTRAL PD	0.600	0.600	0.600	0.0000	0.0000	0.0000
ISR RED TANK PID	0.683	0.743	0.640	-0.0350	-0.0343	-0.0350
ISR MRLS PID	0.595	0.920	0.590	-0.0088	-0.0063	-0.0069
ISR INTEL PID	0.423	0.203	0.402	-0.1460	-0.1488	-0.1505
ISR BLUE TANK PID	0.858	0.900	0.725	-0.0978	-0.0986	-0.0999
ISR NEUTRAL PID	0.410	0.322	0.482	-0.1257	-0.1267	-0.1284
CID RED TANK PID	0.878	0.918	0.875	-0.1340	-0.1343	-0.1363
CID MRLS PID	0.918	0.995	0.989	-0.1828	-0.1821	-0.1851
CID INTEL PID	0.813	0.665	0.807	-0.1877	-0.1909	-0.1933
CID BLUE TANK PID	0.995	0.995	0.974	**-0.0173**	**-0.0173**	**-0.0172**
CID NEUTRAL PID	0.824	0.714	0.827	-0.0023	-0.0023	-0.0023

The results of differentiation in the evaluator's framework can be seen in Table 5.12. The MOPs used to produce the partial differentiation weights are the same set used in the Chapter IV application and can be found in Table A.3. The results hint at the reason for the insensitivity of the evaluator's framework with the MSTAR data (40). The values indicate that the most significant change would result from a deviation in the self-assessment accuracy measures. However, these values were evaluated at their maximum value (as the measures were not available in MSTAR) and could only be decreased. Increasing the probabilistic performance measures, such as P_{ID} and P_{CC}, will result in an increase in the value score, as expected. One interesting result is that each model indicates that increasing the redeployment monetary cost would increase the value score at the particular point in MOP space. Also notice that the partial differentiation values are not merely reproductions of the total possible values attributable to each MOP found in Table 3.1. Though the total possible value results do indicate the *possible* importance placed upon an MOP, the values, much like the tornado diagram results, evaluate one MOP at a time and do not include estimations of the inherent MOP interaction.

Table 5.12 Evaluator Partial Differentiation Vectors at Sample MOP Observations.

		Partial Differentiations					
Objectives	**MOPs**	**CID A**	**ISR A**	**CID B**	**ISR B**	**CID C**	**ISR C**
Robustness	$\%\nabla P_D$	-0.0027	-0.0025	-0.0028	-0.0029	-0.0026	-0.0026
	$\%\nabla P_{ID}$	-0.0018	-0.0017	-0.0017	-0.0017	-0.0017	-0.0018
	$\%\nabla P_{CC}$	-0.0018	-0.0018	-0.0019	-0.0018	-0.0018	-0.0019
Detection Performance	$FAR\|P_D$	**-0.0625**	**-0.0625**	**-0.0625**	**-0.0625**	**-0.0625**	**-0.0625**
	$P_{FA}\|P_D$	**0.0579**	**0.0579**	**0.0579**	**0.0579**	**0.0579**	**0.0579**
Employment Concept		**-0.0300**	**-0.0300**	**-0.0300**	**-0.0300**	**-0.0300**	**-0.0300**
Declaration Ability	P_{DEC}	0.0260	0.0260	0.0260	0.0260	0.0260	0.0260
Classification Ability	P_{ID}	0.0008	0.0005	0.0008	0.0006	0.0008	0.0005
	P_{CC}	0.0011	0.0011	0.0011	0.0011	0.0011	0.0011
Development Cost	Money	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Time	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Expertise	-0.0030	-0.0030	-0.0030	-0.0030	-0.0030	-0.0030
	Risk	-0.0152	-0.0152	-0.0152	-0.0152	-0.0152	-0.0152
Redeployment Cost	Money	0.0244	0.0244	0.0244	0.0244	0.0244	0.0244
	Time	-0.0001	-0.0001	-0.0001	-0.0001	0.0000	0.0000
	Expertise	-0.0003	-0.0003	-0.0003	-0.0003	-0.0003	-0.0003
	Risk	-0.0047	-0.0047	-0.0047	-0.0047	-0.0047	-0.0047
Use Cost	Money	-0.0114	-0.0114	-0.0113	-0.0113	-0.0113	-0.0113
	Time	-0.0168	-0.0168	-0.0168	-0.0168	-0.0168	-0.0168
	Expertise	-0.0280	-0.0280	-0.0280	-0.0280	-0.0280	-0.0280
Self-Assessment	$E_{S\text{-}PD}$	**-0.0840**	**-0.0840**	**-0.0840**	**-0.0840**	**-0.0840**	**-0.0840**
	$E_{S\text{-}PID}$	**-0.1313**	**-0.1313**	**-0.1313**	**-0.1313**	**-0.1313**	**-0.1313**
	$E_{S\text{-}PCC}$	**-0.1747**	**-0.1747**	**-0.1747**	**-0.1747**	**-0.1747**	**-0.1747**

These results may be compared to the tornado diagram approach presented in Section 5.2. The tornado diagrams for the evaluator DA model (Figures 5.3 and 5.4) indicated that only a few of the features offered a significant impact on the overall value when varied: six features for the CID setting and three for the ISR setting. Both settings indicated that value was sensitive to the $E_{S\text{-}PCC}$ measure, the $P_{FA}|P_D$ measure, and the Employment Concept rating. For the CID setting, the three robustness measures showed significant change to the value score when varied. The partial differentiation approach indicate that the features that offer the most change in value (with a partial derivative absolute value greater than or equal to 0.300) are the three self-assessment accuracy measures, the overall detection measures, and the employment concept rating. The highest rated salient feature using this method was the $E_{S\text{-}PCC}$ measure, which is also the most salient measure according to the ISR tornado diagram. Using partial differentiation, the second-most salient feature category was the Overall Detection Performance objective, which includes the $P_{FA}|P_D$ and FAR$|P_D$ measures. The $P_{FA}|P_D$ measure was ranked first in the CID tornado diagram. The two approaches do differ in some respects. The partial differentiation approach considers all of the features in the self-assessment accuracy category as salient while the tornado diagrams make no such indication. While both approaches indicate the direction a feature changes the value, only the tornado diagram indicates whether or not the decision policy changes (color changes). In summary, the calculation of the total possible value attributable to each input is an excellent way to screen for features that should have a large impact on the overall value score. Tornado diagrams are also useful for determining important features, and they indicate when decision policy changes may occur as the input features are varied along their respective domains. Both techniques allow an analyst to rank each input feature by its expected impact on the value score. However, neither of these techniques account for any interactive effects between the features. Determining salient features via partial differentiation does offer a means to account for the inherent interactions within the inputs. This technique provides an analyst insight into the expected value score impact for a change in feature value while any other feature is also varied. The fact that the three techniques produce three different views of the features and their influence on the overall DA model outputs illustrates that analysts have several different tools for sensitivity analysis at their disposal.

VI. Multinomial Selection Procedure

6.1 Introduction

The direct comparison of value and utility scores, like those calculated in Chapter IV, may be misleading because the ATR performance measures that characterize the individual ATR CSs are estimates of the true MOPs. In fact, the MOPs are random variables, and standard procedure is to represent them with their respective sample means. Obviously, this approach ignores the inherent uncertainty surrounding the parameter estimate. Chapter V illustrated how slight changes within the MOPs affect the overall value score results. This chapter illustrates how a multinomial selection procedure (MSP) may be used not only to account for the variability within the estimation of the MOPs, but also to provide a certain level of confidence surrounding the comparison of multiple ATR CSs. By simply using an ordered evaluation measure to distinguish between multiple ATR CSs, the MSP allows the selection of a *best* performer and introduces variation within the MOP estimates, which the Chapter IV comparisons lacked. A variation of the MSP introduced in Chapter II is used in this section.

This chapter details the steps taken to perform the MSP on the ATR evaluation methodology presented. First, assumptions must be made to allow for the use of the MSP. To begin the process, the input MOPs are treated random variables rather than a point estimate. A random draw from each MOP distribution is taken. This draw is propagated through the surrogate combat model to produce a random sample of the value scores. For example, rather than using a value of 0.6 as the probability of detection against Intel vehicles, the P_D is drawn from a normal distribution with a mean of 0.6 and a given standard deviation. The resultant value scores are then compared via the MSP. The performance data of three notional ATR CSs introduced in Chapter IV provides a simple example of the procedure.

6.2 Assumptions

To implement the MSP, certain assumptions need to be made concerning the data. First, larger is assumed to be better. Thus, a higher value or utility score is representative of a better system. Secondly, it is assumed that there is a constant probability of success over all test trials. This assumption holds as long as the test trials are at random, and the probabilities of success obtained are still estimates of the probabilities of winning in any randomly selected trial. Finally, it is assumed that the trials are independent both across and within the systems. This is a reasonable assumption considering the method by which an ATR selects and scores features from an individual region of interest (ROI) within a target scene.

6.3 MOP Estimation

The MOPs of the ATR CSs are based upon estimates collected during ATR testing. For instance, an ATR CS undergoes testing prior to operational use and is proposed to operate at a given performance level, e.g. at a P_D setting of 0.9, an ATR CS may be said to operate at a P_{ID} of 0.5. However, this statistic is based upon randomized test data. Thus, both performance measures, in this case, are also random variables. This uncertainty detracts any confidence in comparisons made between ATR CSs via value scores. To account for this uncertainty, the MOPs used in the MSP are drawn from a normal population with a mean of the estimated performance level and predetermined standard deviation, i.e. P_D~Normal(0.9, 0.05). The standard deviation value was chosen to be small enough to ensure that the performance measures remained reasonably well within the employment concept regions depicted in Figure 4.1. Each of the random draws is also bounded at a value of one since the MOPs in question are probabilities. Drawing MOP value estimates randomly could reflect sampling error of the MOP estimation made during the performance measure assessment of an ATR CS.

6.4 Application

A procedure similar to the one described in formulas 2.45 and 2.46 is implemented in the following application. The procedure is described as:

1. Given *n* test data points, compare estimated value scores for each of the *i* classifiers.

2. Select the best classifier for each data point as the classifier with the maximum estimated value score.

3. Compute the number of wins/successes Y_i for each classifier *i*.

4. Let $Y_{[1]} \leq Y_{[2]} \leq Y_{[3]}$ be the ranked number of successes from Step 3. Select the classifier associated with the largest count, $Y_{[3]}$, as the best.

Thus, the equations used are:

$$\hat{p}_i = \frac{Y_i}{n}, \tag{6.1}$$

and

$$\hat{p}_i \pm Z_{1-\frac{\alpha}{2}} \sqrt{\frac{\hat{p}_i (1-\hat{p}_i)}{n}}, \tag{6.2}$$

where p_i is the probability of system i being the best, $Z_{1-\alpha/2}$ is the test statistic using the normal approximation (large sample size; n>30), and (1-α/2) is the desired level of confidence. Equation 6.1 provides the formula for calculating the probability of being the best performer via a value score point estimate, and equation 6.2 describes the confidence interval generated around the probability of being the best performer, i.e. P_{BEST}.

Using the assumptions listed above, the MSP may be applied to an ATR CS comparison example. The data used for the application in Chapter IV was used and is found in Table A.1. The P_D, P_{ID}, P_{CC}, and P_{FA} values found in the table served as the means of the normal distribution from which the actual MOP estimates used were taken. A standard deviation of 0.05 was selected for each random draw. The randomized MOPs were then input to both the evaluator DA model and the surrogate combat model. The combat model results were then introduced to the warfighter DA model. Both DA models produced the value scores used in the MSP. An analysis on utility scores could be performed similarly, but were not accomplished here. The results of the MSP using the resultant value scores were generated using formulas 2.45 and 2.46 are found in Tables 6.1 and 6.2.

Table 6.1 MSP Results for Evaluator Framework.

Test Data	Value Score			Win/Successes = 1		
Number	ATR A	ATR B	ATR C	ATR A	ATR B	ATR C
1	0.6043	0.6040	0.6404	0	0	1
2	0.5995	0.6039	0.6401	0	0	1
3	0.6007	0.6073	0.6357	0	0	1
…	…	…	…	…	…	…
300	0.5983	0.6066	0.6371	0	0	1
	Successes (Y_i) =			0	0	300
	P_{BEST} = (Y_i/n) =			0.00%	0.00%	100.00%

Table 6.2 MSP Results for Warfighter Framework.

Test Data	Value Score			Win/Successes = 1		
Number	ATR A	ATR B	ATR C	ATR A	ATR B	ATR C
1	0.7756	0.7575	0.4156	1	0	0
2	0.6241	0.3852	0.7949	0	0	1
3	0.7286	0.5950	0.4100	1	0	0
…	…	…	…	…	…	…
300	0.5509	0.8334	0.5376	0	1	0
	Successes (Y_i) =			99	123	78
	P_{BEST} = (Y_i/n) =			33.00%	41.00%	26.00%

Table 6.3 P_{BEST} Confidence Intervals for Evaluator Framework.

P_{BEST}	ATR A	ATR B	ATR C
Upper	0.00%	0.00%	100.00%
Estimate	0.00%	0.00%	100.00%
Lower	0.00%	0.00%	100.00%

Table 6.4 P_{BEST} Confidence Intervals for Warfighter Framework.

P_{BEST}	ATR A	ATR B	ATR C
Upper	38.32%	46.57%	30.96%
Estimate	33.00%	41.00%	26.00%
Lower	27.68%	35.43%	21.04%

6.5 Conclusions

From the MSP results in Tables 6.3 and 6.4, it is clear that the best overall ATR CS from each perspective is easy to distinguish for the evaluator. The procedure provides insight into selecting the better system and a methodology for confidence associated with selecting a best performer. In regards to the ATR evaluator's decision, both ATRs A and B can be eliminated from the comparison, ATR C is the only ATR worth considering. The number of successes indicates that the overall value is not very sensitive to changes made only to the performance areas varied within the evaluator model (P_{ID}, P_{FA}, and P_{CC}). This is reasonable in that previous sensitivity analysis performed upon the evaluator's DA model indicated that no one variable, varied up to 10 percent of its estimated value, significantly altered the value score enough to change the ATR CS selection decision (40). From the warfighter's perspective, ATR B seems to be a better choice, but cannot be considered the best choice as the lower bound of ATR C's P_{BEST} (35.43%) is lower than the P_{BEST} upper bound of ATR CS A (27.68%). The conclusions from Chapter IV indicated that with the given preference structure, the evaluator would prefer ATR C, while the warfighter would prefer ATR B. The MSP results further support these findings, and reflect sensitivity of the value scores to changes within the MOP estimates that were not necessarily noticeable with any associated confidence. Again, examining Table A.2, it is interesting that the ATR with the largest average *AUC* measure (ATR A) is the only one chosen for elimination by both of the interested parties. The results indicate that sensitivity can be very valuable for gaining insight into decision-maker preferences, eliminating competitors from comparison, and making decisions with an associated level of confidence.

VII. Summary and Recommendations

7.1 Overview

This dissertation research provides a methodology for improving programmatic decision-making within the realm of ATR technology development. This section summarizes the resultant contributions of the research and lists possibilities for future research.

7.2 Summary

This dissertation research implemented expanded decision analysis practices to provide an evaluation methodology for ATR CS comparison. The proposed methodology models the subjective preferences of both an ATR evaluator decision-maker and the eventual product user. First, the methodology presents a way to synthesize the many ATR performance measures into two different scores for value and utility, which incorporate the preference structure and risk attitude of the evaluator decision-maker. Next, a methodology is presented that translates the performance characteristics of a particular ATR CS into measures of effectiveness via a combat model. These MOEs are then introduced to the warfighter decision analysis model, which also produces both a value and utility score.

While these scores serve as valuable insight to both parties, there are limitations to direct comparison techniques. The value scores provide little insight as to the sensitivity of the ATR systems to the MOPs. Further, these scores ignore the variation inherent within the MOPs of the ATR CSs. An MSP is introduced to enable selection of the best ATR CS in terms of value. The MSP not only accounts for the variation within the ATR performance measures, but also provides a level of confidence with decisions made concerning the value scores. Following the selection of the best alternative, a sensitivity analysis approach is described. The sensitivity analysis described examines the DA framework inputs by calculating the total possible value attributable to each input, producing tornado diagrams of each input, and calculating salient measures via partial differentiation. Linear regression is used to create a value (utility) response surface to enable sensitivity analysis via differentiation. With these results, evaluators and warfighters may determine the value they place on an individual ATR CS and their respective performance characteristics, which aids in the decision making process throughout the life-cycle of automatic target recognition technology development. Figure 7.1 illustrates the decision analysis approach to a problem and how this dissertation research used these concepts.

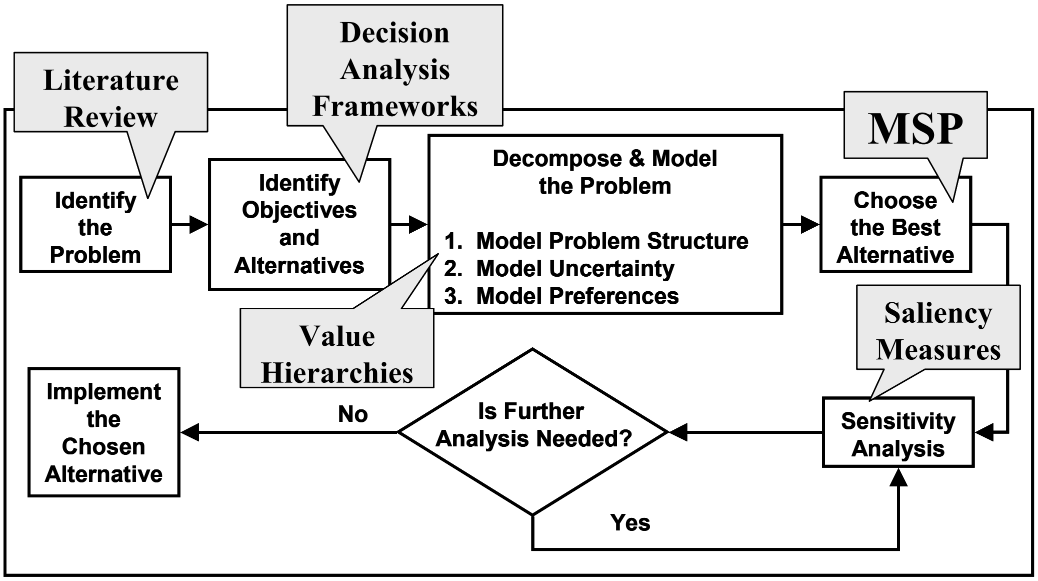

Literature Review
Decision Analysis Frameworks
MSP
Identify the Problem
Identify Objectives and Alternatives
Decompose & Model the Problem
1. Model Problem Structure
2. Model Uncertainty
3. Model Preferences
Choose the Best Alternative
Value Hierarchies
Saliency Measures
Implement the Chosen Alternative
No
Is Further Analysis Needed?
Sensitivity Analysis
Yes

Figure 7.1 Decision Analysis Approach and Implementation.

7.3 Contributions

This section summarizes the contributions resulting from this research.

7.3.1 Development of an Overall Methodology for ATR Technology Evaluation.

The chief contribution that this research offers is an overall approach to evaluating the various technologies being developed under the responsibility of the Sensors Directorate of the Air Force Research Laboratory.

7.3.2 Development of an Evaluator Decision Analysis Framework.

This dissertation presents a methodology for constructing a decision analysis framework for use in ATR CS evaluation, particularly for comparison between competing CSs. This framework is the result of joint dissertation research conducted by the author and Col William K. Klimack, USA. The research detailing the evaluator DA framework has already resulted in a technical report, and supporting results have been included in further dissertation research (11,39,40).

7.3.3 Development of a Warfighter Decision Analysis Framework.

This dissertation introduces a methodology for constructing a decision analysis framework from the warfighter's perspective. This methodology is important since it has been shown that optimizing ATR CS MOPs does not necessarily translate into desirable operational results. The framework allows an ATR evaluator to discern what MOP mixture may produce an optimal mixture for operational results.

7.3.4 Development of a Methodology for MOP to MOE Translation.

This dissertation describes a method of translating MOPs associated with an ATR CS into MOEs that capture the operational results when an ATR technology is applied to an operational environment. It is also noteworthy that the combat model used to demonstrate this methodology is an USAF-accepted model with an impressive VV&A pedigree.

7.3.5 Development of a Heuristic for Determining the Number of Simulation Runs Necessary to Gain a Desired Confidence Interval Half-length about a Value/Utility Score Estimate.

The process, detailed in Chapter V, offers an analyst a technique for determining an acceptable number of runs needed to calculate an acceptable confidence interval width for value or utility score estimate comparison.

7.4 Recommendations

There are many potential avenues of research branching from this research. This section highlights a few of the possibilities.

7.4.1 Sensitivity Analysis on ATR CS Value Across Differing Scenarios.

Since the preference structure of the warfighter change as the operational scenario changes, i.e. a regional, conventional conflict may have different goals than a global, nuclear conflict; it would be advantageous to capture the robustness of given preference structures across several operational scenarios. The results could be used to aid in designing an ATR that is best suited for accomplishing missions in multiple environments.

7.4.2 Creation of a Defined List of MOPs for Current and Future ATR Technologies.

One of the difficulties of comparing current ATR technologies is the differing ‘lingo’ used by the various organizations producing ATR technologies. For instance, the performance measure labeled PD may mean probability of detection to one researcher, but probability of declaration to another. Creating a single vocabulary for all ATR technology developers, evaluators, and users would strengthen the argument for incorporating the decision analysis frameworks and simplify the MOP and MOE definition process.

7.4.3 Include the Effects of Sensor Fusion.

The idea of making decisions based upon multiple ATR sensors is not a new one, but incorporating the effects of fusing ATR systems (rather than having them compete) could yield interesting results, especially when a value score is associated with the fused systems.

Appendix A. ATR Application Performance Data.

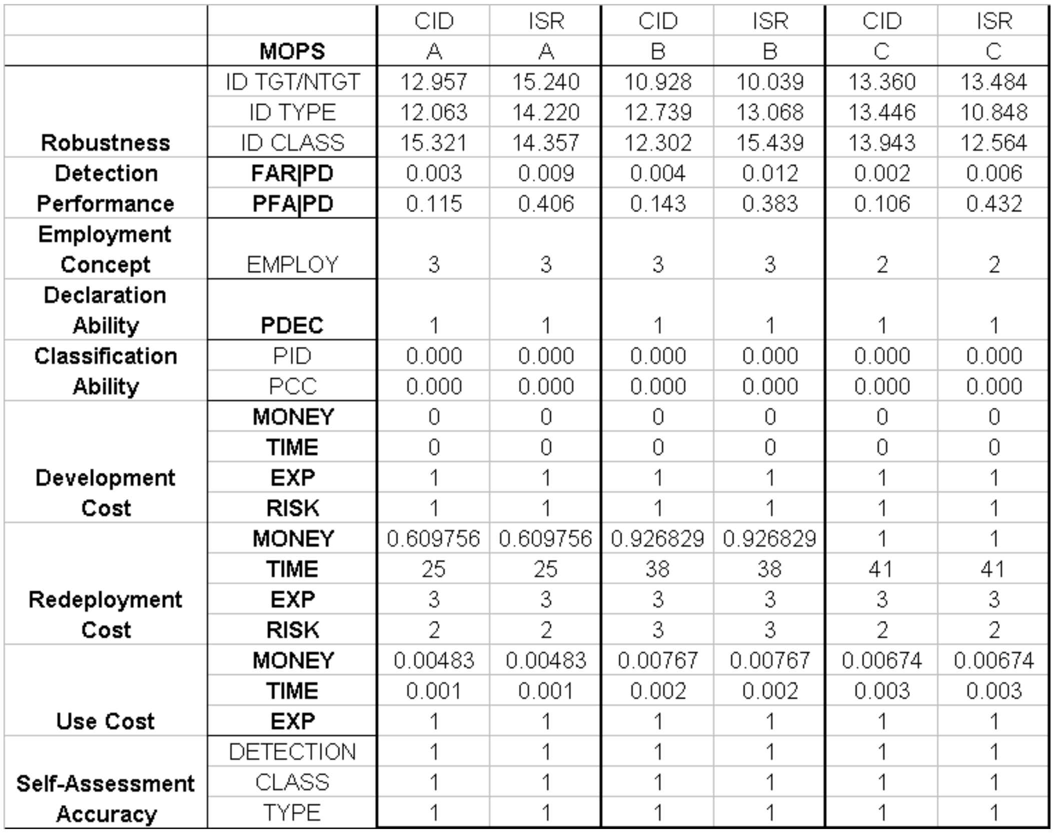
Table A.1 ATR Performance Data.

		CID			ISR		
		A	B	C	A	B	C
P_D	RED_TANK	0.6	0.6	0.6	0.9	0.9	0.9
	MRLS	0.6	0.6	0.6	0.9	0.9	0.9
	INTEL	0.6	0.6	0.6	0.9	0.9	0.9
	BLUE_TANK	0.6	0.6	0.6	0.9	0.9	0.9
	NEUTRAL	0.6	0.6	0.6	0.9	0.9	0.9
	OVERALL	0.6	0.6	0.6	0.9	0.9	0.9
P_{CC}	TANK	0.949	0.970	0.970	0.897	0.925	0.891
	MML	0.960	0.970	0.950	0.760	0.780	0.810
	TRUCK	0.984	0.960	0.977	0.911	0.895	0.888
	OVERALL	0.965	0.966	**0.969**	0.875	**0.884**	0.873
P_{ID}	RED_TANK	0.890	0.960	0.920	0.830	0.840	0.700
	MRLS	0.960	0.970	0.950	0.760	0.780	0.810
	INTEL	0.980	0.910	0.940	0.820	0.700	0.690
	BLUE_TANK	0.870	0.900	0.940	0.690	0.810	0.790
	NEUTRAL	0.950	0.920	0.960	0.800	0.850	0.830
	OVERALL	0.930	0.932	**0.942**	0.780	**0.796**	0.764
P_{FID}	RED_TANK	0.123	0.083	0.125	0.318	0.258	0.360
	MRLS	0.082	0.005	0.011	0.405	0.080	0.410
	INTEL	0.188	0.336	0.193	0.577	0.797	0.598
	BLUE_TANK	0.005	0.005	0.026	0.143	0.100	0.275
	NEUTRAL	0.176	0.286	0.173	0.590	0.678	0.518
P_{FA}	OVERALL	0.115	0.143	**0.106**	0.406	**0.383**	0.432
FAR	# per 10000 km2	3	4	**2**	9	12	**6**
% Change P_D	RED_TANK	7.094	18.156	6.271	18.720	15.389	11.895
	MRLS	16.011	5.931	12.537	19.078	7.500	15.006
	INTEL	15.269	5.864	10.113	12.040	8.797	9.593
	BLUE_TANK	13.734	7.934	19.195	14.839	7.827	16.787
	NEUTRAL	12.678	16.757	18.684	11.520	10.684	14.142
	OVERALL	12.957	**10.928**	13.360	15.240	**10.039**	13.484
% Change P_{ID}	RED_TANK	17.675	9.771	19.323	8.903	11.660	10.549
	MRLS	7.372	17.239	12.513	19.898	9.742	14.920
	INTEL	7.909	17.602	8.574	15.271	11.410	8.831
	BLUE_TANK	19.538	10.324	15.709	15.766	17.152	11.035
	NEUTRAL	7.821	8.759	11.111	11.264	15.374	8.904
	OVERALL	**12.063**	12.739	13.446	**14.220**	13.068	**10.848**
% Change P_{CC}	RED_TANK	11.282	9.351	15.253	18.815	15.760	11.453
	MRLS	13.714	16.499	17.443	10.638	14.775	11.114
	INTEL	15.556	8.768	15.876	15.150	18.105	16.372
	BLUE_TANK	18.136	18.833	12.559	14.785	16.350	15.769
	NEUTRAL	17.917	8.061	8.584	12.397	12.206	8.112
	OVERALL	15.321	**12.302**	13.943	14.357	15.439	**12.564**

Table A.2 ATR Binormal ROC Data.

			CID					ISR	
	TGT Type	**A**	**B**	**C**		**TGT Type**	**A**	**B**	**C**
Distance	**RED_TANK**	2.40	3.15	2.56		**RED_TANK**	1.43	1.65	0.88
Between	**MRS**	3.15	4.46	3.97		**MRS**	0.95	2.18	1.11
Means	**INTEL**	2.95	1.76	2.43		**INTEL**	0.73	-0.30	0.28
(Δm)	**BLUE_TANK**	3.71	3.86	3.51		**BLUE_TANK**	1.57	2.16	1.40
	NEUTRAL	2.58	1.97	2.69		**NEUTRAL**	0.61	0.58	0.90
AUC	**RED_TANK**	0.9552	0.9870	0.9649		**RED_TANK**	0.8440	0.8783	0.7331
Measure	**MRS**	0.9870	0.9992	0.9975		**MRS**	0.7491	0.9384	0.7837
	INTEL	0.9815	0.8933	0.9571		**INTEL**	0.6971	0.4160	0.5785
	BLUE_TANK	0.9956	0.9956	0.9935		**BLUE_TANK**	0.8665	0.9367	0.8389
	NEUTRAL	0.9659	0.9182	0.9714		**NEUTRAL**	0.6669	0.6591	0.7377
Average		0.9770	0.9587	0.9769			0.7647	0.7657	0.7344
(Rank)		(1)	(3)	(2)			(2)	(1)	(3)

Table A.3 MOP Inputs for Evaluator Framework.



		CID	ISR	CID	ISR	CID	ISR
	MOPS	A	A	B	B	C	C
	ID TGT/NTGT	12.957	15.240	10.928	10.039	13.360	13.484
	ID TYPE	12.063	14.220	12.739	13.068	13.446	10.848
Robustness	ID CLASS	15.321	14.357	12.302	15.439	13.943	12.564
Detection	**FAR|PD**	0.003	0.009	0.004	0.012	0.002	0.006
Performance	**PFA|PD**	0.115	0.406	0.143	0.383	0.106	0.432
Employment Concept	EMPLOY	3	3	3	3	2	2
Declaration Ability	**PDEC**	1	1	1	1	1	1
Classification	PID	0.000	0.000	0.000	0.000	0.000	0.000
Ability	PCC	0.000	0.000	0.000	0.000	0.000	0.000
	MONEY	0	0	0	0	0	0
	TIME	0	0	0	0	0	0
Development	**EXP**	1	1	1	1	1	1
Cost	**RISK**	1	1	1	1	1	1
	MONEY	0.609756	0.609756	0.926829	0.926829	1	1
	TIME	25	25	38	38	41	41
Redeployment	**EXP**	3	3	3	3	3	3
Cost	**RISK**	2	2	3	3	2	2
	MONEY	0.00483	0.00483	0.00767	0.00767	0.00674	0.00674
	TIME	0.001	0.001	0.002	0.002	0.003	0.003
Use Cost	**EXP**	1	1	1	1	1	1
	DETECTION	1	1	1	1	1	1
Self-Assessment	CLASS	1	1	1	1	1	1
Accuracy	TYPE	1	1	1	1	1	1

Table A.4 MOP to Value and Utility Translation Vectors.

	Value	Utility
INTERCEPT	1.7000	1.7609
ISR RED_TANK PD	0	0
ISR MRS PD	0	0
ISR INTEL PD	0	0
ISR BLUE_TANK PD	0.1095	0.1061
ISR NEUTRAL PD	0	0
CID RED_TANK PD	-0.1537	-0.1723
CID MRS PD	-0.2613	-0.2690
CID INTEL PD	-0.1961	-0.1995
CID BLUE_TANK PD	0	0
CID NEUTRAL PD	0	0
ISR RED_TANK PID	-0.0508	-0.0638
ISR MRS PID	-0.1323	-0.1331
ISR INTEL PID	0	0
ISR BLUE_TANK PID	-0.0914	-0.0933
ISR NEUTRAL PID	-0.1144	-0.1185
CID RED_TANK PID	-0.1218	-0.1367
CID MRS PID	-0.1755	-0.1796
CID INTEL PID	-0.1723	-0.1741
CID BLUE_TANK PID	0	0
CID NEUTRAL PID	0	0

Table A.5 MOP to MOE Translation Matrix, *T*.

	Pct of Dumb Bombs Remaining	Pct of WMD Remaining	Pct of S/S Missiles Remaining	Pct of A/A & S/A Remaining	Pct Red Forces Damaged	Pct of Red Personnel Killed	Pct Red C2 Damaged	Length of Battle	Number of Dead Civilians	Number of Damaged Neutrals	Number of Fratricide Occurrences	Pct Blue Weapons Remaining	Pct of Blue Personnel Remaining	Pct of Blue C2 Remaining	Pct Remaining Dumb Bombs	Pct Remaining Precision Bombs	Pct Remaining CMs	Pct Remaining S/A
INTERCEPT	0	0	-0.9930	0	2.2403	1.9418	2.4881	58.7070	5.3685	2.4772	2.8760	0.8963	1.0335	0	0.8807	0.8774	0	0
ISR RED_TANK PD	0	0	0.2329	0	0	0	0	0	0	0	1.6875	0	0	0	0	0	0	0
ISR MRS PD	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ISR INTEL PD	0	0	0	0	0	0	-0.1961	0	0	0	-1.1460	0.0403	-0.0200	0	0	0	0	0
ISR BLUE_TANK PD	0	0	0	0	0.0991	0.0856	0.1814	3.6250	3.3128	0	0	0	0	0	-0.0079	-0.0094	0	0
ISR NEUTRAL PD	0	0	0	0	0	0	0	0	2.2543	0	0	0	0	0	0	0	0	0
CID RED_TANK PD	0	0	0.3876	0	-0.2536	-0.2198	-0.1764	-6.9870	-1.9376	-0.5904	0	0	0.0060	0	0.0163	0.0144	0	0
CID MRS PD	0	0	0.0593	0	-0.4785	-0.4148	-0.2917	-12.9928	-3.3653	-0.8958	-0.6543	0.0469	0.0134	0	0.0271	0.0270	0	0
CID INTEL PD	0	0	0	0	-0.1143	-0.0991	-0.4375	-3.2352	0	0	0	-0.013167	0.0068	0	0.0059	0.0062	0	0
CID BLUE_TANK PD	0	0	0	0	0	0	0	0	0	0	-0.38475	0	0	0	-0.0030	0.0039	0	0
CID NEUTRAL PD	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ISR RED_TANK PID	0	0	0.2637	0	-0.1009	-0.0874	0	-2.75833	-0.873583	0	-0.3737	0.0129	0	0	0.0061	0.0053	0	0
ISR MRS PID	0	0	-0.0394	0	-0.1728	-0.1497	0.1306	-4.5399	0	0	0	0	0	0	0.0095	0.0095	0	0
ISR INTEL PID	0	0	0	0	-0.0495	-0.0430	-0.3209	-1.2362	0	0	0	0	0	0	0.0035	0	0	0
ISR BLUE_TANK PID	0	0	0.0379	0	-0.0980	-0.0850	-0.1576	-3.0427	-2.33475	-0.6624	0	0	0	0	0.0079	0.0049	0	0
ISR NEUTRAL PID	0	0	0.0557	0	-0.1330	-0.1152	-0.1991	-3.5446	0	0	0	0	0.0134	0	0.0075	0.0092	0	0
CID RED_TANK PID	0	0	0.2842	0	-0.1899	-0.1645	-0.1704	-5.1648	-0.9181	0	-0.4509	0.0246	0	0	0.0108	0.0109	0	0
CID MRS PID	0	0	0.0525	0	-0.3276	-0.2840	-0.1718	-9.0437	-2.6503	-0.7574	0	0.0179	0.0116	0	0.0169	0.0199	0	0
CID INTEL PID	0	0	0	0	-0.0852	-0.0737	-0.4040	-2.3383	-0.9212	0	-0.5521	0.0274	0	0	0.0052	0.0041	0	0
CID BLUE_TANK PID	0	0	0	0	0	0	0	0	2.0312	0.4866	0.9152	-0.0908	-0.0666	0	-0.0042	0	0	0
CID NEUTRAL PID	0	0	0	0	0	0	0	1.1140	3.2931	0.8437	0.3556	0	0	0	0	0	0	0

Table A.6 Fractional Factorial Design Matrix.

			PD										PID									
			ISR					CID					ISR					CID				
Obs	Block	Pattern	RED TANK	MRLS	INTEL	BLUE TANK	NEUTRAL	RED TANK	MRLS	INTEL	BLUE TANK	NEUTRAL	RED TANK	MRLS	INTEL	BLUE TANK	NEUTRAL	RED TANK	MRLS	INTEL	BLUE TANK	NEUTRAL
1	2	------++-+--++--+-++	0.9	0.9	0.9	0.9	0.9	0.6	0.75	0.75	0.6	0.75	0.7	0.7	0.85	0.85	0.7	0.85	0.99	0.85	0.99	0.99
2	2	----++--+-++--++-+--	0.9	0.9	0.9	0.9	0.95	0.75	0.6	0.6	0.75	0.6	0.85	0.85	0.7	0.7	0.85	0.99	0.85	0.99	0.85	0.85
3	1	---+-+--+-++-+--+-++	0.9	0.9	0.9	0.95	0.9	0.75	0.6	0.6	0.75	0.6	0.85	0.85	0.7	0.85	0.7	0.85	0.99	0.85	0.99	0.99
4	1	---++-++-+--+-++-+--	0.9	0.9	0.9	0.95	0.95	0.6	0.75	0.75	0.6	0.75	0.7	0.7	0.85	0.7	0.85	0.99	0.85	0.99	0.85	0.85
5	1	--+--+--++--+-++--++	0.9	0.9	0.95	0.9	0.9	0.75	0.6	0.6	0.75	0.75	0.7	0.7	0.85	0.7	0.85	0.99	0.85	0.85	0.99	0.99
6	1	--+-+-++--++-+--++--	0.9	0.9	0.95	0.9	0.95	0.6	0.75	0.75	0.6	0.6	0.85	0.85	0.7	0.85	0.7	0.85	0.99	0.99	0.85	0.85
7	2	--++--++--++--++--++	0.9	0.9	0.95	0.95	0.9	0.6	0.75	0.75	0.6	0.6	0.85	0.85	0.7	0.7	0.85	0.99	0.85	0.85	0.99	0.99
8	2	--++++--++--++--++--	0.9	0.9	0.95	0.95	0.95	0.75	0.6	0.6	0.75	0.75	0.7	0.7	0.85	0.85	0.7	0.85	0.99	0.99	0.85	0.85
9	1	-+---+-+--+-+-+-++-+	0.9	0.95	0.9	0.9	0.9	0.75	0.6	0.75	0.6	0.6	0.85	0.7	0.85	0.7	0.85	0.85	0.99	0.99	0.85	0.99
10	1	-+--+-+-++-+-+-+--+-	0.9	0.95	0.9	0.9	0.95	0.6	0.75	0.6	0.75	0.75	0.7	0.85	0.7	0.85	0.7	0.99	0.85	0.85	0.99	0.85
11	2	-+-+--+-++-+--+-++-+	0.9	0.95	0.9	0.95	0.9	0.6	0.75	0.6	0.75	0.75	0.7	0.85	0.7	0.7	0.85	0.85	0.99	0.99	0.85	0.99
12	2	-+-+++-+--+-++-+--+-	0.9	0.95	0.9	0.95	0.95	0.75	0.6	0.75	0.6	0.6	0.85	0.7	0.85	0.85	0.7	0.99	0.85	0.85	0.99	0.85
13	2	-++---+-+-+-++-+-+-+	0.9	0.95	0.95	0.9	0.9	0.6	0.75	0.6	0.75	0.6	0.85	0.7	0.85	0.85	0.7	0.99	0.85	0.99	0.85	0.99
14	2	-++-++-+-+-+--+-+-+-	0.9	0.95	0.95	0.9	0.95	0.75	0.6	0.75	0.6	0.75	0.7	0.85	0.7	0.7	0.85	0.85	0.99	0.85	0.99	0.85
15	1	-+++-+-+-+-+-+-+-+-+	0.9	0.95	0.95	0.95	0.9	0.75	0.6	0.75	0.6	0.75	0.7	0.85	0.7	0.85	0.7	0.99	0.85	0.99	0.85	0.99
16	1	-++++-+-+-+-+-+-+-+-	0.9	0.95	0.95	0.95	0.95	0.6	0.75	0.6	0.75	0.6	0.85	0.7	0.85	0.7	0.85	0.85	0.99	0.85	0.99	0.85
17	1	+----++----++--++++-	0.95	0.9	0.9	0.9	0.9	0.75	0.75	0.6	0.6	0.6	0.7	0.85	0.85	0.7	0.7	0.99	0.99	0.99	0.99	0.85
18	1	+---+--++++--++----+	0.95	0.9	0.9	0.9	0.95	0.6	0.6	0.75	0.75	0.75	0.85	0.7	0.7	0.85	0.85	0.85	0.85	0.85	0.85	0.99
19	2	+--+---++++----++++-	0.95	0.9	0.9	0.95	0.9	0.6	0.6	0.75	0.75	0.75	0.85	0.7	0.7	0.7	0.7	0.99	0.99	0.99	0.99	0.85
20	2	+--++++----++++----+	0.95	0.9	0.9	0.95	0.95	0.75	0.75	0.6	0.6	0.6	0.7	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.99
21	2	+-+----++--++++--++-	0.95	0.9	0.95	0.9	0.9	0.6	0.6	0.75	0.75	0.6	0.7	0.85	0.85	0.85	0.85	0.85	0.85	0.99	0.99	0.85
22	2	+-+-+++--++----++--+	0.95	0.9	0.95	0.9	0.95	0.75	0.75	0.6	0.6	0.75	0.85	0.7	0.7	0.7	0.7	0.99	0.99	0.85	0.85	0.99
23	1	+-++-++--++--++--++-	0.95	0.9	0.95	0.95	0.9	0.75	0.75	0.6	0.6	0.75	0.85	0.7	0.7	0.85	0.85	0.85	0.85	0.99	0.99	0.85
24	1	+-+++--++--++--++--+	0.95	0.9	0.95	0.95	0.95	0.6	0.6	0.75	0.75	0.6	0.7	0.85	0.85	0.7	0.7	0.99	0.99	0.85	0.85	0.99
25	2	++-------++++++++---	0.95	0.95	0.9	0.9	0.9	0.6	0.6	0.6	0.6	0.75	0.85	0.85	0.85	0.85	0.85	0.99	0.99	0.85	0.85	0.85
26	2	++--+++++--------+++	0.95	0.95	0.9	0.9	0.95	0.75	0.75	0.75	0.75	0.6	0.7	0.7	0.7	0.7	0.7	0.85	0.85	0.99	0.99	0.99
27	1	++-+-++++----++++---	0.95	0.95	0.9	0.95	0.9	0.75	0.75	0.75	0.75	0.6	0.7	0.7	0.7	0.85	0.85	0.99	0.99	0.85	0.85	0.85
28	1	++-++----++++----+++	0.95	0.95	0.9	0.95	0.95	0.6	0.6	0.6	0.6	0.75	0.85	0.85	0.85	0.7	0.7	0.85	0.85	0.99	0.99	0.99
29	1	+++--++++++++-------	0.95	0.95	0.95	0.9	0.9	0.75	0.75	0.75	0.75	0.75	0.85	0.85	0.85	0.7	0.7	0.85	0.85	0.85	0.85	0.85
30	1	+++-+--------+++++++	0.95	0.95	0.95	0.9	0.95	0.6	0.6	0.6	0.6	0.6	0.7	0.7	0.7	0.85	0.85	0.99	0.99	0.99	0.99	0.99
31	2	++++----------------	0.95	0.95	0.95	0.95	0.9	0.6	0.6	0.6	0.6	0.6	0.7	0.7	0.7	0.7	0.7	0.85	0.85	0.85	0.85	0.85
32	2	++++++++++++++++++++	0.95	0.95	0.95	0.95	0.95	0.75	0.75	0.75	0.75	0.75	0.85	0.85	0.85	0.85	0.85	0.99	0.99	0.99	0.99	0.99

Appendix B. Surrogate Model MATLAB$^{®}$ Subroutines.

SUBROUTINE ATO

```
% author:  Capt Brian Bassham
% date:    24 Apr 02
% Revised: 30 Apr 02

j = 1;
for i = 1:Total_TGTs
   if TGT_List(i,6) ~= 4 & TGT_List(i,6) ~= 5 & TGT_List(i,5) ~= 0 &
TGT_List(i,9) < 1.0   %If the TGT is NOT CLASSIFIED as a
Blue_Tank/Neutral, IS DETECTED and NOT DEAD
      Strike_List(j,1) = TGT_List(i,6);   % What the ISR thinks it is
      Strike_List(j,2) = TGT_List(i,1);   % What it actually is
      Strike_List(j,3) = TGT_List(i,2);   % Its X position
      Strike_List(j,4) = TGT_List(i,3);   % Its Y position
      Strike_List(j,5) = TGT_List(i,8);   % Its target number
      j = j + 1;
   end
end

Strike_List = sortrows(Strike_List,1);
%Sort the list based upon the ISR's classification
%Note: EADSIM2 assumes a priority: Red_Tank, MRS, Intel.
```

SUBROUTINE BDA

```
function [New]=BDA(a,b,N,M)

if b == 1
    for j = 1:N
        if M(j,8) == a
            M(j,9) = rand(1) + M(j,9);
        end
    end
end


if b == 2
    for j = 1:N
        if M(j,8) == a
            M(j,9) = 1;
        end
    end
end

New = M;
```

SUBROUTINE CID

```
% author:  Capt Brian Bassham
% date:    24 Apr 02
% Revised: 30 APR 02

Fighter_Size = min(Num_CID,size(Strike_List,1));

for i = 1:Fighter_Size
   Fighter(i,1:4)    = Strike_List(i,1:4);
   Fighter(i,5:7)    = [0     0     0];
   Fighter(i,8:9)    = [0.5   0.05];
   Fighter(i,10)     = Strike_List(i,5);
end

%Calculate the Range of the CID platform to the TGT
%for i = 1:Fighter_Size
%   Fighter(i,5) = sqrt((Fighter(i,3)-Fighter(i,8))^2+(TGT_List(i,4)-
Fighter(i,9))^2);
%end

%Start the CID strike loop
%[1] Determine if the CID radar can detect the TGT
for i = 1:Fighter_Size
    rn = rand(1);
    if rn < PD_CID(Fighter(i,2))        %Determine if the CID platform
detects the TGT (Use actual ID to determine result)
        Fighter(i,6) = 1;
    else
        Fighter(i,6) = 0;
    end
%    if Fighter(i,4) <= 0.25 | TGT_List(i,4) >= 0.95
%      Fighter(i,5) = 0;        %The detection setting is overwritten
if out of range
%   end

%[2] Classify the detected TGT
    if Fighter(i,6) == 1
      rn = rand(1);
      if rn <= PID_CID(Fighter(i,2),1)          %Base result upon
actual ID
         Fighter(i,7) = 1;
      elseif rn <= PID_CID(Fighter(i,2),2)
         Fighter(i,7) = 2;
      elseif rn <= PID_CID(Fighter(i,2),3)
         Fighter(i,7) = 3;
      elseif rn <= PID_CID(Fighter(i,2),4)
         Fighter(i,7) = 4;
      else
         Fighter(i,7) = 5;
      end
    end

    %[3] Attempt to destroy TGT, if the CID and ISR classifications
agree
```

```
    if Fighter(i,7) == Fighter(i,1)
        %Determine if the bomb is a dumb bomb or precision-guided
munition
        rn3 = rand(1);
        if rn3 < 0.5
           Damage_Distance = Dumb_Damage_Distance;
           Dumb_Bombs = Dumb_Bombs + 1;
        else
           Damage_Distance = Precision_Damage_Distance;
           Precision_Bombs = Precision_Bombs + 1;
        end
        % The CID platform fires
        rn = rand(1);
        Weapons_Expended = Weapons_Expended + 1;
        %Incidental Damage Loop
        for h=1:Total_TGTs
            if (TGT_List(h,2) > (Fighter(i,3)- Damage_Distance)) &
(TGT_List(h,2) < (Fighter(i,3)+ Damage_Distance))      %if TGT is in X-
range
                if (TGT_List(h,3) > (Fighter(i,4)- Damage_Distance)) &
(TGT_List(h,3) < (Fighter(i,4)+ Damage_Distance))      %if TGT is in Y-
range
                    if TGT_List(h,9) < 1.0
%if TGT is alive
                        if  TGT_List(h,8) ~= Fighter(i,10)
%if TGT is not original TGT
                            TGT_List(h,9) = rand(1);
                        end
                    end
                end
            end
        end

        if rn < Damage
          if Fighter(i,2) < Friendly_Level
              Hostiles_Damaged = Hostiles_Damaged + 1;
              TGT_List = BDA(Fighter(i,10),1,Total_TGTs,TGT_List);
          elseif Fighter(i,2) > Friendly_Level
              Neutrals_Damaged = Neutrals_Damaged + 1;
              TGT_List = BDA(Fighter(i,10),2,Total_TGTs,TGT_List);
          else
              Allies_Damaged = Allies_Damaged + 1;
              TGT_List = BDA(Fighter(i,10),1,Total_TGTs,TGT_List);
          end

        elseif rn < Kill
          if Fighter(i,2) < Friendly_Level
              Hostiles_Killed = Hostiles_Killed + 1;
              TGT_List = BDA(Fighter(i,10),2,Total_TGTs,TGT_List);
          elseif Fighter(i,2) > Friendly_Level
              Neutrals_Killed = Neutrals_Killed + 1;
              TGT_List = BDA(Fighter(i,10),2,Total_TGTs,TGT_List);
          else
              Frat = Frat + 1;
              TGT_List = BDA(Fighter(i,10),2,Total_TGTs,TGT_List);
          end

        else
          Weapon_Miss = Weapon_Miss + 1;
        end
    end
end
```

SUBROUTINE EADSIM2

```
function [Value,MOEsM,MOPs]=  EADSIM2(A,B,C,D,E,F)
% EADSIM2 is a set of Matlab subroutines designed to mimic the effects
of ATR technology in an
% operational environment as executed in the EADSIM combat model.  This
model is created for
% support of Capt Bassham's dissertation research on a new ATR CS
evaluation methodology.
% author:  Capt Brian Bassham
% date:    24 Apr 02
% Revised: 16 May 02

clear TGT_List;clear Strike_List;clear Fighter;clear Damaged_TGTs

%MOPs of the ATR in Question
%Probabilities of Detection
PD_ISR   = [0.92  0.92  0.92  0.92     0.92];
PD_CID      = [0.65     0.65  0.65  0.65     0.65];

%Probabilities of Identification/Misidentification
%Based upon: [1] Red_Tank [2] MSR [3] Intel [4] Blue_Tank [5] Neutral
Object
%PID for ISR must be > 0.7
%PID for CID must be > 0.95
PID_ISR     = [   0.80  0.82  0.87  0.97     1.0;
                  0.03  0.83  0.87  0.92     1.0;
                  0.04  0.09  0.89  0.92     1.0;
               0.12  0.13  0.15  0.95     1.0;
                  0.03  0.05  0.15  0.20     1.0];
PID_CID     = [   0.92  0.93  0.95  0.99     1.0;
                  0.02  0.94  0.97  0.99     1.0;
                  0.01  0.02  0.94  0.96     1.0;
               0.05  0.06  0.07  0.99     1.0;
                    0.01   0.02  0.07  0.08     1.0];

PD_ISRM = [mean(PD_ISR(1:3)) PD_ISR(4) PD_ISR(5)];
PD_CIDM = [mean(PD_CID(1:3)) PD_CID(4) PD_CID(5)];
PID_ISRM = [mean([PID_ISR(1,1) PID_ISR(2,2)-PID_ISR(2,1) PID_ISR(3,3)-
PID_ISR(3,2)])  PID_ISR(4,4)-PID_ISR(4,3)  PID_ISR(5,5)-PID_ISR(5,4)];
PID_CIDM = [mean([PID_CID(1,1) PID_CID(2,2)-PID_CID(2,1) PID_CID(3,3)-
PID_CID(3,2)])  PID_CID(4,4)-PID_CID(4,3)  PID_CID(5,5)-PID_CID(5,4)];
MOPs = [PD_ISRM PID_ISRM PD_CIDM PID_CIDM]';

%False Alarm Rate
FAR = 0.0007;
```

```
%Combat Model Outcome Percentages
Kill           = 0.75;        %Friend's ability to strike a TGT
Damage         = 0.23;        %Friend's ability to damage a TGT
Dumb_Damage_Distance = 0.035;     %TGTs within a certain radius of dumb
bombs are damaged
Precision_Damage_Distance = 0.015;     %TGTs within a certain radius of
precision bombs are damaged
Dumb_Bombs = 0;
Precision_Bombs = 0;

%Force Initialization (Based upon User inputs):
http://www.army.mil/CMH-PG/books/www/Wwindx.htm
Num_Red_Tank   = A;        %Assume crew of 5   50 mph    >3000 m
   [T-72]
Num_MRS        = B;            %Assume crew of 5   46 mph   >20,380 m
   (40)  [BM-21]
Num_Intel      = C;        %Assume crew of 5   65 mph
   [Hummer or M1974]
Num_Blue_Tank  = D;        %Assume crew of 4   41 mph    >3000 m
   [M1A1]
Num_Neutral    = E;            %Assume crew of 2   50 mph
Total_TGTs     = Num_Red_Tank + Num_MRS + Num_Intel + Num_Blue_Tank +
Num_Neutral;
Red_Forces     = A + B;
Red_C2         = C;
Red_Total      = Red_Forces + Red_C2;
Num_CID        = F;       %The number of Allied fighters available for
vectoring
Friendly_Level = 4;  %The number at which TGTs not to be attacked begin
(Blue_Tanks & Neutrals are not to be hit)

%MOE Initialization
Frat              = 0;
Allies_Damaged    = 0;
Hostiles_Killed   = 0;
Hostiles_Damaged  = 0;
Neutrals_Damaged  = 0;
Neutrals_Killed   = 0;
Weapons_Expended  = 0;
Weapon_Miss       = 0;
Red_Tank_Damaged    = 0;
Red_Tank_Dead       = 0;
MRS_Damaged = 0;
MRS_Dead = 0;
Intel_Damaged = 0;
Intel_Dead = 0;
Blue_Tank_Damaged = 0;
Blue_Tank_Dead = 0;
Neutral_Damaged = 0;
Neutral_Dead = 0;


%Create the target list
%The nine columns are:
```

```
%[TGT_Type      X Position Y Position  Range_to_ISR   Detected?
   ID_Type?  Last_ID_Type       TGT #       Dead/Damaged ]
TGT_List = zeros(Total_TGTs,9);

%Create the Air Tasking Order (ATO)
%The five columns are:
%[ID_Type   Actual_Type    X Position     Y Position  TGT #]
Strike_List = zeros(1,5);

%Create the Fighter's target list
%The 10 columns are:
%[Classified_Type    Actual_Type    TGT_X_Position    TGT_Y_Position
Range_to_CID   Detected?   Classified As  X_Pos_Fighter  Y_Pos_Fighter
TGT #]
Fighter = zeros(Num_CID,10);

% Begin Combat Model Execution
Scenario    %Places objects in the scene and generates a starting map

Loops = 10;

%while ((Red_Tank_Damaged + Red_Tank_Dead + MRS_Damaged + MRS_Dead +
Intel_Damaged + Intel_Dead)/Red_Total) < 1.0
for l=1:Loops
%   ((Red_Tank_Damaged + Red_Tank_Dead + MRS_Damaged + MRS_Dead +
Intel_Damaged + Intel_Dead)/Red_Total)
   ISR         %Simulates the ISR platform detecting/classifying
targets & calls ATO (Strike List generation) & CID (Vectors fighters)
modules
   Stats
%   Loops = Loops + 1;
end
Stats
[Value,H]=warfighter_curves(MOEs,1);
```

SUBROUTINE ISR

```
% author:  Capt Brian Bassham
% date:    24 Apr 02
% Revised: 01 May 02

% This initializes the band width of the ISR platform's (SAR)
visibility
% The band will stretch across the entire map but will be limited to
the
% Y-boundaries between 0.25 and 0.8.  Thus, targets above and below
this band will
% not be detected unless they move into it or the ISR platform moves
closer to/further away.

%Run through each TGT to see if it can be detected and Classified
%Update the TGT List
for i = 1:size(TGT_List,1)
   %Calculate Range to ISR Platform
   TGT_List(i,4) = sqrt((TGT_List(i,2)-XPos_Blue_ISR)^2+(TGT_List(i,3)-
YPos_Blue_ISR)^2);
   %The Last ID Type is set as the previously detected ID
   TGT_List(i,7) = TGT_List(i,6);
   % Can the TGT be detected?  Based upon the PD value of the ISR
sensor
   rn = rand(1);
   if rn <= PD_ISR(TGT_List(i,1))
        TGT_List(i,5) = 1;
   else
      TGT_List(i,5) = 0;
      TGT_List(i,6) = 0;      %If not detected this turn then the
Classified ID is set back to nothing
   end
   % The TGT cannot be detected if outside of the range of the ISR
Platform (overwrites previous result)
   if TGT_List(i,4) <= 0.0 | TGT_List(i,4) >= 1.95
      TGT_List(i,5) = 0;        %The detection setting is overwritten
if out of range
   end
   % Identify TGT as...based upon PID value of the ISR sensor
   if TGT_List(i,5)==1      %If the TGT was detected
      rn = rand(1);
      if rn <= PID_ISR(TGT_List(i,1),1)
         TGT_List(i,6) = 1;
      elseif rn <= PID_ISR(TGT_List(i,1),2)
         TGT_List(i,6) = 2;
      elseif rn <= PID_ISR(TGT_List(i,1),3)
         TGT_List(i,6) = 3;
      elseif rn <= PID_ISR(TGT_List(i,1),4)
         TGT_List(i,6) = 4;
      else
         TGT_List(i,6) = 5;
      end
   end
   %Assign a TGT Number
   TGT_List(i,8) = i;
end

TGT_List = sortrows(TGT_List,4);       %Randomize the entries so that
the Red_Tanks are not always chosen first.

ATO      %Calls the subroutine that generates the ATO
CID      %Calls the subroutine that allows the Fighters to strike their
TGTs

%Change the ISR Platform's position
XPos_Blue_ISR = (0.2+0.6*rand(1));
```

SUBROUTINE MAP

```
% The map is assumed to be a unit square with (0,0) being the lower
left corner
% author:  Capt Brian Bassham
% date:    25 Apr 02
% Revised:

%This plots the objects in their original starting places within the
scenario
figure(1)
plot(TGT_List(1:Level_1,2),TGT_List(1:Level_1,3),'rv')
hold on
plot(TGT_List(Level_1+1:Level_2,2),TGT_List(Level_1+1:Level_2,3),'rd')
hold on
plot(TGT_List(Level_2+1:Level_3,2),TGT_List(Level_2+1:Level_3,3),'rs')
hold on
plot(TGT_List(Level_3+1:Level_4,2),TGT_List(Level_3+1:Level_4,3),'b^')
hold on
plot(TGT_List(Level_4+1:Level_5,2),TGT_List(Level_4+1:Level_5,3),'gs')
hold on
plot(XPos_Blue_Airfield,YPos_Blue_Airfield,'bx',XPos_Blue_ISR,YPos_Blue
_ISR,'bo')
hold on
axis([0 1 0 1])
hold off
```

SUBROUTINE MAP2

```
% The map is assumed to be a unit square with (0,0) being the lower
left corner
% author:  Capt Brian Bassham
% date:    25 Apr 02
% Revised:

TGT_List=sortrows(TGT_List,1);

%This plots the objects and includes destoyed/damaged objects
figure(2)
plot(TGT_List(1:Level_1,2),TGT_List(1:Level_1,3),'rv')
hold on
plot(TGT_List(Level_1+1:Level_2,2),TGT_List(Level_1+1:Level_2,3),'rd')
hold on
plot(TGT_List(Level_2+1:Level_3,2),TGT_List(Level_2+1:Level_3,3),'rs')
hold on
plot(TGT_List(Level_3+1:Level_4,2),TGT_List(Level_3+1:Level_4,3),'b^')
hold on
plot(TGT_List(Level_4+1:Level_5,2),TGT_List(Level_4+1:Level_5,3),'gs')
hold on
plot(Damaged_TGTs(:,2),Damaged_TGTs(:,3),'kp')
hold on
plot(XPos_Blue_Airfield,YPos_Blue_Airfield,'bx',XPos_Blue_ISR,YPos_Blue
_ISR,'bo')
hold on
axis([0 1 0 1])
hold off
```

SUBROUTINE SCENARIO

```
% This subroutine positions the objects within the scenario
% The map is assumed to be a unit square with (0,0) being the lower
left corner
% author:  Capt Brian Bassham
% date:    24 Apr 02
% Revised: 25 Apr 02

%Variables for ease of programming
Level_1 = Num_Red_Tank;
Level_2 = Num_Red_Tank + Num_MRS;
Level_3 = Num_Red_Tank + Num_MRS + Num_Intel;
Level_4 = Num_Red_Tank + Num_MRS + Num_Intel + Num_Blue_Tank;
Level_5 = Num_Red_Tank + Num_MRS + Num_Intel + Num_Blue_Tank +
Num_Neutral;

%This places Red_Tanks in a box with X boundaries at 0.2-0.8 & Y
boundaries between 0.5 & 0.7
for i = 1: Level_1
   TGT_List(i,1) = 1;
   TGT_List(i,2) = (0.2+0.8*rand(1));
   TGT_List(i,3) = (0.5+0.2*rand(1));
end
%This places MRSs in a box with X boundaries at 0.4-0.9 & Y boundaries
between 0.5 & 0.75
for i = Level_1 + 1:Level_2
   TGT_List(i,1) = 2;
   TGT_List(i,2) = (0.4+0.5*rand(1));
   TGT_List(i,3) = (0.5+0.25*rand(1));
end
%This places Intel Trucks in a box with X boundaries at 0.5-0.95 & Y
boundaries between 0.65 & 0.9
for i = Level_2 + 1:Level_3
   TGT_List(i,1) = 3;
   TGT_List(i,2) = (0.5+0.45*rand(1));
   TGT_List(i,3) = (0.65+0.25*rand(1));
end
%This places Blue_Tanks in a box with X boundaries at 0.2-0.8 & Y
boundaries between 0.2 & 0.6
for i = Level_3 + 1:Level_4
   TGT_List(i,1) = 4;
   TGT_List(i,2) = (0.2+0.6*rand(1));
   TGT_List(i,3) = (0.2+0.4*rand(1));
end
%This places Neutral Objects in a box with X boundaries at 0.3-0.9 & Y
boundaries between 0.3 & 0.8
for i = Level_4+ 1:Level_5
   TGT_List(i,1) = 5;
   TGT_List(i,2) = (0.3+0.6*rand(1));
   TGT_List(i,3) = (0.3+0.5*rand(1));
end

%This places the Blue_Airfield at the following X-Y Coordinates
XPos_Blue_Airfield = 0.5;
YPos_Blue_Airfield = 0.05;
```

```
%This places the Blue_ISR at the following X-Y Coordinates
XPos_Blue_ISR = (0.2+0.6*rand(1));
YPos_Blue_ISR = 0.1;

%MAP
```

SUBROUTINE STATS

```
% author:  Capt Brian Bassham
% date:    25 Apr 02
% Revised: 14 May 02

Weapon_Info = [Weapons_Expended  Weapon_Miss];

FEN_Damaged_Killed = [Allies_Damaged  Frat  Hostiles_Damaged
Hostiles_Killed Neutrals_Damaged  Neutrals_Killed];

Damaged_TGTs=zeros(1,9);

j=1;
for k = 1:Total_TGTs
   if TGT_List(k,9) > 0.0
      Damaged_TGTs(j,:) = TGT_List(k,:);
      j = j+1;
   end
end

%Damaged_TGTs;
Incidental_Allied_Damage   = 0;
Incidental_Allied_Kill     = 0;
Incidental_Hostile_Damage  = 0;
Incidental_Hostile_Kill    = 0;
Incidental_Neutral_Damage  = 0;
Incidental_Neutral_Kill    = 0;
Red_Tank_Damaged    = 0;
Red_Tank_Dead       = 0;
MRS_Damaged = 0;
MRS_Dead = 0;
Intel_Damaged = 0;
Intel_Dead = 0;
Blue_Tank_Damaged = 0;
Blue_Tank_Dead = 0;
Neutral_Damaged = 0;
Neutral_Dead = 0;

%The loop which calculates proximity kills/hits
for k=1:j-1
   if Damaged_TGTs(k,1) < Friendly_Level & Damaged_TGTs(k,9) < 1.0
      Incidental_Hostile_Damage = Incidental_Hostile_Damage + 1;
   elseif Damaged_TGTs(k,1) < Friendly_Level & Damaged_TGTs(k,9) >= 1.0
      Incidental_Hostile_Kill = Incidental_Hostile_Kill + 1;
   elseif Damaged_TGTs(k,1) == Friendly_Level & Damaged_TGTs(k,9) < 1.0
      Incidental_Allied_Damage = Incidental_Allied_Damage + 1;
   elseif Damaged_TGTs(k,1) == Friendly_Level & Damaged_TGTs(k,9) >=
1.0
      Incidental_Allied_Kill = Incidental_Allied_Kill + 1;
   elseif Damaged_TGTs(k,1) > Friendly_Level & Damaged_TGTs(k,9) < 1.0
      Incidental_Neutral_Damage = Incidental_Neutral_Damage + 1;
   else
      Incidental_Neutral_Damage = Incidental_Neutral_Damage + 1;
   end
end

%The loop which calculates the
for k=1:j-1
   if Damaged_TGTs(k,1) == 1 & Damaged_TGTs(k,9) < 1.0
      Red_Tank_Damaged = Red_Tank_Damaged + 1;
   elseif Damaged_TGTs(k,1) == 1 & Damaged_TGTs(k,9) >= 1.0
      Red_Tank_Dead = Red_Tank_Dead + 1;
   elseif Damaged_TGTs(k,1) == 2 & Damaged_TGTs(k,9) < 1.0
      MRS_Damaged = MRS_Damaged + 1;
   elseif Damaged_TGTs(k,1) == 2 & Damaged_TGTs(k,9) >= 1.0
      MRS_Dead = MRS_Dead + 1;
   elseif Damaged_TGTs(k,1) == 3 & Damaged_TGTs(k,9) < 1.0
      Intel_Damaged = Intel_Damaged + 1;
   elseif Damaged_TGTs(k,1) == 3 & Damaged_TGTs(k,9) >= 1.0
      Intel_Dead = Intel_Dead + 1;
   elseif Damaged_TGTs(k,1) == 4 & Damaged_TGTs(k,9) < 1.0
      Blue_Tank_Damaged = Blue_Tank_Damaged + 1;
   elseif Damaged_TGTs(k,1) == 4 & Damaged_TGTs(k,9) >= 1.0
      Blue_Tank_Dead = Blue_Tank_Dead + 1;
   elseif Damaged_TGTs(k,1) == 5 & Damaged_TGTs(k,9) < 1.0
      Neutral_Damaged = Neutral_Damaged + 1;
   else
      Neutral_Dead = Neutral_Dead + 1;
   end
end
Incidental = [Incidental_Allied_Damage  Incidental_Allied_Kill
Incidental_Hostile_Damage Incidental_Hostile_Kill
Incidental_Neutral_Damage Incidental_Neutral_Kill];
Total_Damage = [Red_Tank_Damaged  Red_Tank_Dead  MRS_Damaged  MRS_Dead
Intel_Damaged  Intel_Dead  Blue_Tank_Damaged  Blue_Tank_Dead
Neutral_Damaged  Neutral_Dead];

% [1]  Pct Remaining Dumb Bombs       [2]  Pct Remaining WMD
[3]  Pct Remaining CMs & S/S         [4]  Pct Remaining A/A & S/A
% [5]  Pct Red Systems Damaged        [6]  Pct Red Personnel Destroyed
[7]  Pct Red C2 Destoyed              [8]  Empty
% [9]  Length of Battle               [10] # of Dead Civilians
[11] # of Neutral Objects Destroyed  [12] # of Fratricide occurrences
% [13] Pct Blue Systems Remaining     [14] Pct Blue Personnel Remaining
[15] Pct Blue C2 Remaining           [16] Empty
% [17]  Pct Remaining Dumb Bombs      [18] Pct Remaining Precision
Bombs [19] Pct Remaining CMs & S/S         [20] Pct Remaining A/A & S/A

MOEs = [ 0 0 1-(MRS_Dead/B) 0;
```

```
   (Red_Tank_Damaged+Red_Tank_Dead+MRS_Damaged+MRS_Dead)/Red_Forces
(5*(Red_Tank_Dead+MRS_Dead+Intel_Dead))/(5*Red_Total)  (Intel_Damaged +
Intel_Dead)/Red_C2   0;
   Loops     Neutral_Damaged+Neutral_Dead*(round(rand(1)+1))
Neutral_Damaged+Neutral_Dead  Frat;
   1-((Blue_Tank_Damaged+Blue_Tank_Dead)/D)  ((Num_Blue_Tank*5)-
(Blue_Tank_Dead*5))/(Num_Blue_Tank*5)   0 0;
   1-(Dumb_Bombs/(Loops*Num_CID*0.5))   1-
(Precision_Bombs/(Loops*Num_CID*0.5)) 0 0];

MOEsM = [0 0 1-(MRS_Dead/B) 0
(Red_Tank_Damaged+Red_Tank_Dead+MRS_Damaged+MRS_Dead)/Red_Forces
(5*(Red_Tank_Dead+MRS_Dead+Intel_Dead))/(5*Red_Total)  (Intel_Damaged +
Intel_Dead)/Red_C2 Loops     Neutral_Damaged+Neutral_Dead*2
Neutral_Damaged+Neutral_Dead  Frat 1-
((Blue_Tank_Damaged+Blue_Tank_Dead)/D)  ((Num_Blue_Tank*5)-
(Blue_Tank_Dead*5))/(Num_Blue_Tank*5)   0 1-
(Dumb_Bombs/(Loops*Num_CID*0.5))   1-
(Precision_Bombs/(Loops*Num_CID*0.5)) 0 0]';

% [1]  Pct Remaining Dumb Bombs   [2]  Pct Remaining Precision Bombs
[3]  Pct of Weapons Expended           [4]  Pct of Expended Weapons
that Miss
% [5]  Number of Damaged Neutrals [6]  Number of Dead Neutrals
[7]  Estimate of Dead Civilians        [8]  Length of Battle
% [9]  Pct Blue Weapons Remaining [10] Number of Damaged/Dead Blue Wpns
[11] Pct of Blue Personnel Remaining   [12] # of Fratricide occurrences
% [13] Pct Red Forces Damaged     [14] Pct Red C2 Damaged
[15] Pct of S/S Missiles Remaining     [16] Pct of Red Personnel Killed

%MOEs = [1-(Dumb_Bombs/(Loops*Num_CID*0.5))  1-
(Precision_Bombs/(Loops*Num_CID*0.5))
Weapons_Expended/(Loops*Num_CID)    Weapon_Miss/Weapons_Expended;
%        Neutral_Damaged      Neutral_Dead
Neutral_Damaged+Neutral_Dead*2    Loops;
%        1-((Blue_Tank_Damaged+Blue_Tank_Dead)/D)
   Blue_Tank_Dead+Blue_Tank_Damaged   ((Num_Blue_Tank*5)-
(Blue_Tank_Dead*5))/(Num_Blue_Tank*5)   Frat;
%
(Red_Tank_Damaged+Red_Tank_Dead+MRS_Damaged+MRS_Dead)/Red_Forces
(Intel_Damaged + Intel_Dead)/Red_C2    1-(MRS_Dead/B)    (5*(Red_Total-
(Red_Tank_Dead+MRS_Dead+Intel_Dead)))/(5*Red_Total)];
%MAP2
```

Appendix C. Glossary of Acronyms and Abbreviations.

2AFC	two-alternative, forced-choice
ACC	Air Combat Command
AFRL	Air Force Research Laboratory
AGRI	Air-to-Ground Radar Imaging
ATO	Air Tasking Order
ATR	Automatic Target Recognition
AUC	area under the curve
AVC	All Vector Comparison
BDA	Battle Damage Assessment
BEM	Bechhofer, Elmaghraby, and Morse
CA	classification accuracy
CI	confidence interval
CID	Combat Identification
COMPASE	Comprehensive ATR Scientific Evaluation
CS	Classification System
DA	Decision Analysis
DOE	design of experiments
EADSIM	Extended Air Defense Simulation
EUROC	Expected Utility Receiver Operating Characteristic
FEN	friend/enemy/neutral
FN	false negative
FP	false positive

FROC	Frequency Receiver Operating Characteristic
GUI	graphical user interface
IFF	Identification Friend or Foe
ISR	Intelligence/Surveillance/Reconnaissance
K-S	Komolgorov-Smirnov
LGP	Linear Goal Program/Programming
LROC	Localization Receiver Operating Characteristic
MBT	Main Battle Tank
MOE	Measure of Effectiveness
MOP	Measure of Performance
MRLS	Mobile Rocket Launcher System
MSP	Multinominal Selection Procedure, Multinomial Selection Problem
MSTAR	Moving and Stationary Target Acquisition and Recognition
NCTI	Non-cooperative Target Identification
NCTR	Non-cooperative Target Recognition
N-N	binormal
PDF	probability distribution function
RAC	Response Analysis Characteristic
ROC	Receiver Operating Characteristic
ROI	region of interest
SAR	synthetic aperture radar
SME	subject matter expert
TGT	target

TN	true negative
TP	true positive
USAF	United States Air Force
VV&A	Verification, Validation, and Accreditation

Bibliography

1. Adroit Systems, Inc. *Targets Under Trees (TUT): Constructive Simulation Kickoff Meeting*. Dayton OH, 9 May 2002.

2. Air Force Research Laboratory COMPASE Center. *COMPASE Vocabulary 9/15/00.* n. pag., https://restricted.compase.vdl-atr.afrl.af.mil/foundation/tools/references/vocabulary/vocabulary.html.

3. -----. *ATR Performance Metrics.* n. pag., https://restricted.compase.vdl.afrl.af.mil/agri/acc_summer_2000/metrics.html

4. -----. *MSTAR System Demonstration Evaluation Test Report 8 December 1999.* n. pag., https://restricted.compase.vdl.afrl.af.mil/mstar

5. -----. *SAR ATR Algorithm Comparison 12 September 2000.* n. pag., https://restricted.compase.vdl.afrl.af.mil/mstar

6. -----. *State of the Art Briefing.* n. pag., https://restricted.compase.vdl.afrl.af.mil/soa/products.asp

7. Alsing, S.G. *The Evaluation of Competing Classifiers*. Ph.D. dissertation, Air Force Institute of Technology, Wright-Patterson AFB OH, 2000.

8. Alsing, S.G., K.W. Bauer, and J.O. Miller. "Evaluating Competing Classifiers Using a Multinomial Selection Procedure," *Proceedings of Artificial Neural Networks in Engineering (ANNIE) 2000 Conference*, St. Louis MO, November 5-8, 2000.

9. Alsing, S.G., K.W. Bauer, and J.O. Miller. "A Multinomial Selection Procedure for Evaluating Pattern Recognition Algorithms," *Pattern Recognition*, accepted for publication 9 July 2001.

10. Banks, J., J.S. Carson, and B.L. Nelson. *Discrete-Event System Simulation*. Prentice Hall, Upper Saddle River NJ, 1996.

11. Bassham, C.B., W.K. Klimack, and K.W. Bauer. "ATR evaluation through the synthesis of multiple performance measures," In *Proceedings of the International Society for Optical Engineering (SPIE),* Vol 4729 (Orlando FL, Apr 2002), In print.

12. Bechhofer, R.E., T.J. Santner, and D.M. Goldsman. *Design and Analysis of Experiments for Statistical Selection, Screening, and Multiple Comparisons*. John Wiley and Sons, New York NY, 1995.

13. Campell, G. "Advances in Statistical Methods for the Evaluation of Diagnostic and Laboratory Tests," *Statistics in Medicine,* 13 (1994), 499-508.

14. Cantor, S.B., and M.W. Kattan. "Determining the area under the ROC Curve for a binary diagnostic test," *Medical Decision Making,* 20 (2000), 468-471.

15. Catlin, A.E., K.W. Bauer, E.F. Mykytka, and D.C. Montgomery. "System Comparison Procedures for Automatic Target Recognition Systems," *Naval Research Logistics,* 46 (1999), 357-371.

16. Catlin, A.E., L. Myers, K.W. Bauer, S.K. Rogers, and R. Boussard. "Performance Modeling for Automatic Target Recognition Systems," *Proceedings of the International Society for Optical Engineering (SPIE),*3070 (Orlando FL, Apr 1997), 185-193.

17. Centor, R.M., and J.S. Schwartz. "An evaluation of methods for estimating the area under the receiver operating characteristic (ROC) curve," *Medical Decision Making,* 5 (1982), 149-156.

18. Chankong, V., and Y.Y. Haimes. *Multiobjective Decision Making: Theory and Methodology*. North-Holland, New York NY, 1983.

19. Clemen, R.T. *Making Hard Decisions: An Introduction to Decision Analysis*. Duxbury Press, Belmont CA, 1990.

20. Committee of Technology for Future Naval Forces. *Technology for the United States Navy and Marine Corps, 2000-2035: Becoming a 21st-Century Force*. Volume 2, Technology. http://www.nap.edu/html/tech_21st/t1.htm. Washington DC: Naval Academy Press, 1997.

21. Dreiseitl, S., L. Ohno-Machado, and M. Binder. "Comparing three-class diagnostic tests by three-way ROC analysis," *Medical Decision Making,* 20 (2000), 323-331.

22. Efron, B., and R.J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall, New York NY, 1993.

23. Egan, J.P. *Signal Detection Theory and ROC Analysis*. Academic Press, New York NY, 1975.

24. Feuchter, C.A. *Air Force Analyst's Handbook: On Understanding the Nature of Analysis*. Office of Aerospace Studies, Kirtland NM, Jan 2000.

25. Friedland, D.J., A.S. Go, J.B. Davoren, M.S. Shlipak, S.W. Bent, L.L. Subak, and T. Mendelson. *Evidence-Based Medicine: A Framework for Clinical Practice*. Appleton and Lange, Stamford CT, 1998.

26. Goicoechea, A., D.R. Hansen, and L. Duckstein. *Multiobjective Decision Analysis with Engineering and Business Applications*. John Wiley and Sons, New York NY, 1982.

27. Green, D.M. and J.A. Swets. *Signal Detection Theory and Psychophysics*. John Wiley and Sons Inc., New York NY, 1966.

28. Gujarati, D.N. *Basic Econometrics*. McGraw-Hill, Inc., New York NY, 1988.

29. Hall, D.L. *Mathematical Techniques in MultisensorData Fusion*. Artech House, Boston MA, 1992.

30. Han, R.Y., and R.L. Clark. "Characterization and evaluation of automatic target recognizer performance," *Proceedings of the International Society for Optical Engineering* 504 (Bellingham, 1984), 341-351.

31. Hand, D.J. *Construction and Assessment of Classification Rules*. John Wiley and Sons, New York NY, 1997.

32. Hanley, J.A. "The use of the 'Binormal' model for parametric ROC analysis of quantitative diagnostic tests," *Statistics in Medicine,* 15 (1996), 1575-1585.

33. Harvey, R.G., K.W. Bauer, and J.R. Litko. "Constrained System Optimization and Capability Based Analysis", *Military Operations Research* Vol 2 Number 4 (1996), 5-19.

34. Heagerty, P.J., T. Lumley, and M.S. Pepe. "Time-Dependent ROC Curves for Censored Survival Data and a Diagnostic Marker," *Biometrics,* 56 (2000), 337-344.

35. Higdon, J.M. *Utility of Experimental Design in Automatic Target Recognition Performance Evaluation*. M.S. Thesis, Air Force Institute of Technology, Wright-Patterson AFB OH, 2001.

36. Hilden, J. "The area under the ROC curve and its competitors," *Medical Decision Making,* 11 (1991), 95-101.

37. Jekel, J.F., J.G. Elmore, and D.L. Katz. *Epidemiology, Biostatistics, and Preventive Medicine*. W.B. Saunders Company, Philadelphia PA, 1996.

38. Kester, A.D., and F. Buntinx. "Meta-analysis of ROC curves," *Medical Decision Making,* 20 (2000), 430-439.

39. Klimack, W.K. *Robustness of Multiple Objective Decision Analysis Functions*. Ph.D. dissertation, Air Force Institute of Technology, Wright-Patterson AFB OH, 2002.

40. Klimack, W.K., C.B. Bassham, and K.W. Bauer. *Application of Decision Analysis to Automatic Target Recognition Programmatic Decisions*, Technical Report AFIT/EN-TR-02-06, Air Force Institute of Technology, Wright-Patterson AFB OH Apr 2002.

41. Law, A.M., and W.D. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, Inc., New York, NY, 1991.

42. Leonard, J. AFRL/SNAA, Wright-Patterson AFB OH. Personal Interview and Personal Correspondence. 26 June 2002.

43. McClish, D.K. "Comparing the areas under more than two independent ROC curves," *Medical Decision Making,* 7 (1987), 149-155.

44. -----. "Combining and comparing area estimates across studies or strata," *Medical Decision Making,* 12 (1992), 274-279.

45. Miller, J.O., B.L. Nelson, and C.H. Reilly, "Estimating the Probability That a Simulated System Will Be the Best," *Naval Research Logisitcs*, 49 (2002), 341-358.

46. Moons, K.G., T. Stijnen, B.C. Michel, H.R. Buller, G.A. Van Es, D.E. Grobbee, and J.D. Habbema. "Application of treatment thresholds to diagnostic-test evaluation: an alternative to the comparison of areas under receiver operating characteristic curves," *Medical Decision Making,* 17 (1997), 447-454.

47. Mossman, D. "Three-way ROCs," *Medical Decision Making,* 19 (1999), 78-89.

48. Myers, R.H., and D.C. Montgomery. *Response Surface Methodology*. John Wiley & Sons, Inc., New York NY, 1995.

49. Nielson, G., H. Hagen, and H. Muller. *Scientific Visualization:Overviews, Methodologies, and Techniques*. IEEE Computer Society, Los Alamitos CA, 1997.

50. Pepe, M.S. "Three Approaches to Regression Analysis of Receiver Operating Characteristic Curves for Continuous Test Results," *Biometrics,* 54 (1998), 124-135.

51. Philosophov, L.V. "On ROC Analysis," *Medical Decision Making,* 16 (1996), 302-303.

52. Raubertas, R.F., L.E. Rodewald, S.G. Humiston, and P.G. Szilagyi. "ROC Curves and Classification Trees," *Medical Decision Making,* 14 (1984), 169-174.

53. Richter, K., and W.R. Steinbach. "Multiple classification and receiver operating characteristic (ROC) analysis," *Medical Decision Making,* 7 (1987), 234-237.

54. Riegelman, R.K. *Studying a Study and Testing a Test*. Lippincott, Williams, & Wilkins, Philadelphia PA, 4th ed, 2000.

55. Ross, T.D. AFRL/SNAT, Wright-Patterson AFB OH. Personal Interviews. 9 February 2001, 23 February 2001, and 12 April 2001.

56. Ross, T. D., J.J. Bradley, L.J. Hudson, and M.P. O'Connor. "SAR ATR – So What's the Problem? - An MSTAR Perspective," *Proceedings of the International Society for Optical Engineering (SPIE)*, vol. 3721.

57. Ross, T. D., and J.C. Mossing. "The MSTAR Evaluation Methodology," *Proceedings of the International Society for Optical Engineering (SPIE)*, vol. 3721.

58. Ross, T. D., L.A. Westerkamp, E. G. Zelino, and T.L. Burns. "Extensibility and other model-based ATR evaluation concepts," *Proceedings of the International Society for Optical Engineering (SPIE),* 3070 (Orlando FL, Apr 1997), 213-222.

59. Sadowski, C. ACC/DRSA, Langley AFB, VA. Personal Interview and Personal Correspondence. 27 August 2001, 9 October 2001, 19 October 2001, 21 February 2002, 5 March 2002, and 13 June 2002.

60. Sadowski, C. *Measuring Combat Identification…A Warfighter Perspective.* Automatic Target Recognition Theory Workshop, Wright State University, Dayton OH, 8-9 August 2001.

61. SAS Institute, Inc. *JMP IN: Statistics Made Visual Help Guide.* SAS Institute, Inc., Cary NC, 1997.

62. Schniederjans, M.J. *Goal Programming: Methodology and Applications*. Kluwer Academic Publishers, Boston MA, 1995.

63. -----. *Linear Goal Programming*. Petrocelli Books, Inc., Princeton NJ, 1984.

64. Schubert, F.N., and T.L. Kraus. *The Whirlwind War: The United States Army in Operations Desert Shield and Desert Storm*. U.S. Army Center or Military History, Washington DC, 1995.

65. Sheskin, D.J. *Handbook of Parametric and Nonparametric Statistical Procedures*. Chapman and Hall/CRC, Boca Raton FL, 2000.

66. Swets, J.A., and R.M. Pickett. *Evaluation of Diagnostic Systems: Methods from Signal Detection Theory*. Academic Press, New York NY, 1982.

67. Teledyne Brown Engineering, Inc. *Extended Air Defense Simulation (EADSIM) Executive Summary*. n. pag., http://www.eadsim.com/EADSIMExecSum.pdf. Teledyne Brown Engineering Information Technology, Huntsville AL, Nov 2000.

68. -----. *Extended Air Defense Simulation (EADSIM) Methodology Manual*. Teledyne Brown Engineering Information Technology, Huntsville AL, Nov 2000.

69. Washburn, A.R. *Search and Detection*. Military Applications Section – Operations Research Society of America, Arlington VA, May, 1981.

70. Zhou, X.H. "Empirical Bayes combination of estimated areas under ROC curves using estimating equations," *Medical Decision Making,* 16 (1996), 24-28.

Vita

Captain Christopher Brian Bassham was born in Columbia, Tennessee. He graduated from Richland High School (Lynnville, Tennessee) as the Co-Valedictorian in 1991 and attended the United States Air Force Academy in Colorado Springs, Colorado. In 1995, Captain Bassham graduated from the Air Force Academy with a Bachelor of Science degree in Operations Research and was commissioned into the United States Air Force. His first assignment was to Barksdale Air Force Base in Louisiana, where he served as a flight test engineer for testing of conventional and nuclear air-launched cruise missiles. In August 1997, Captain Bassham entered the School of Engineering at the Air Force Institute of Technology (AFIT) to pursue a Master of Science degree in Operational Analysis, which he achieved in March 1999. Captain Bassham remained at the School of Engineering to pursue a Ph.D. in Operations Research. Upon graduation, he will be assigned to Scott AFB, IL.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to an penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* 12-09-2002	**2. REPORT TYPE** **Dissertation**	**3. DATES COVERED** *(From – To)* MAR 1999-AUG 2002

4. TITLE AND SUBTITLE
AUTOMATIC TARGET RECOGNITION CLASSIFICATION SYSTEM EVALUATION METHODOLOGY

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Christopher B. Bassham, Capt, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)
AFIT/ENS
2950 P St, Building 640
Dept. of Operational Sciences
Air Force Institute of Technology
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/DS/ENS/02-03

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/ Sensors Directorate ATTN: James D. Leonard
2241 Avionics Cl Building 620 Rm N3-X19 WPAFB, OH 45433

ACC/DRSA (Combat ID) ATTN: Maj Stewart DeVilbiss
204 Dodd Blvd, Langley AFB, VA 23665

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This dissertation research makes contributions towards the evaluation of developing Automatic Target Recognition (ATR) technologies through the application of decision analysis (DA) techniques. ATR technology development decisions should rely not only on the measures of performance (MOPs) associated with a given ATR classification system (CS), but also on the expected measures of effectiveness (MOEs). The purpose of this research is to improve the decision-making process for ATR technology development. A decision analysis framework that allows decision-makers in the ATR community to synthesize the performance measures, costs, and characteristics of each ATR system with the preferences and values of both the evaluators and the warfighters is developed. The inclusion of the warfighter's perspective is important in that it has been proven that basing ATR CS comparisons solely upon performance characteristics does not ensure superior operational effectiveness. The methodology also captures the relationship between MOPs and MOEs via a combat model. An example scenario demonstrates how ATR CSs may be compared. Sensitivity analysis is performed to demonstrate the robustness of the MOP to value score and MOP to MOE translations. A multinomial selection procedure is introduced to account for the random nature of the MOP estimates.

15. SUBJECT TERMS
Automatic Target Recognition, multinomial selection procedure, measures of performance, measures of effectiveness, Decision Analysis, comparison, value, utility, ROC curve, evaluation

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Dr. Kenneth W. Bauer, Jr., Professor
a. REPORT	b. ABSTRACT	c. THIS PAGE			**19b. TELEPHONE NUMBER** *(Include area code)*
U	U	U	UU	227	(937) 255-6565, ext 4328