NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

EVALUATION OF FRAUD DETECTION DATA MINING USED IN THE AUDITING PROCESS OF THE DEFENSE FINANCE AND ACCOUNTING SERVICE

by

Donald J. Jenkins

June 2002

Thesis Advisor: Samuel E. Buttrey
Second Reader Lyn R. Whitaker

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, 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 this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2002	**3. REPORT TYPE AND DATES COVERED** Master's Thesis
4. TITLE AND SUBTITLE: Evaluation of Fraud Detection Data Mining Used in the Auditing Process of the Defense Finance and Accounting Service		**5. FUNDING NUMBERS**
6. AUTHOR (S) Jenkins, Donald J.		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)		**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the U.S. Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE**

13. ABSTRACT *(maximum 200 words)*

The Defense Finance and Accounting Service (DFAS) uses data mining to analyze millions of vendor transactions each year in an effort to combat fraud. The long timeline required to investigate potential fraud precludes DFAS from using fraud as a supervised modeling performance measure, so instead it uses the conditions needing improvement (CNI) found during site audits. To verify this method, a thorough literature review is conducted which demonstrates a clear relationship between fraud and CNIs. Then recent site audits are analyzed to prove that supervised modeling is detecting CNIs at a higher rate than random record selection. The next phase of the research evaluates recent models to determine if models are improving with each new audit. Finally, to enhance the supervised modeling process, four initiatives are proposed: a revised model scoring implementation, a knowledge base of audit results, alternative model streams for record selection and a recommended modeling process for the CNI knowledge base. The goal of the proposed enhancements is to improve an already successful program so that the data-mining efforts will further reduce taxpayer losses through fraud, error or misappropriation of funds.

14. SUBJECT TERMS: Fraud Detection, Data Mining, Auditing, Neural Networks, Classification			**15. NUMBER OF PAGES** 128
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 **Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. 239-18**

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

EVALUATION OF FRAUD DETECTION DATA MINING USED IN THE AUDITING PROCESS OF THE DEFENSE FINANCE AND ACCOUNTING SERVICE

Donald J. Jenkins
Lieutenant, United States Navy
B.A./B.S., University of San Diego, 1994

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
June 2002

Author: Donald J. Jenkins

Approved by: Samuel E. Buttrey
Thesis Advisor

Lyn R. Whitaker
Second Reader

James Eagle
Chairman, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The Defense Finance and Accounting Service (DFAS) uses data mining to analyze millions of vendor transactions each year in an effort to combat fraud. The long timeline required to investigate potential fraud precludes DFAS from using fraud as a supervised modeling performance measure, so instead it uses the conditions needing improvement (CNI) found during site audits. To verify this method, a thorough literature review is conducted which demonstrates a clear relationship between fraud and CNIs. Then recent site audits are analyzed to prove that supervised modeling is detecting CNIs at a higher rate than random record selection. The next phase of the research evaluates recent models to determine if models are improving with each new audit. Finally, to enhance the supervised modeling process, four initiatives are proposed: a revised model scoring implementation, a knowledge base of audit results, alternative model streams for record selection and a recommended modeling process for the CNI knowledge base. The goal of the proposed enhancements is to improve an already successful program so that the data-mining efforts will further reduce taxpayer losses through fraud, error or misappropriation of funds.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION 1
A. PURPOSE. 1
B. BACKGROUND. 1
C. RESEARCH QUESTIONS. 2
D. SCOPE OF THESIS. 3
E. METHODOLOGY. 4
F. ORGANIZATION OF STUDY. 4
G. EXPECTED BENEFITS OF THIS THESIS. 6

II. BACKGROUND 7
A. BACKGROUND INFORMATION. 7
1. Description of DFAS IR Seaside. 7
2. Literature Review. 8
B. DESCRIPTION OF IR SEASIDE'S DATA MINING. 10
1. Overview of IR Seaside Analytical Procedures. 10
a. Duplicate Payments. 11
b. Supervised Modeling. 11
c. Unsupervised Modeling. 12
d. Related Payments. 12
e. Random Records. 12
2. In-depth Review of Supervised Modeling. 13
a. Description of Fraud Knowledge Base. ... 14
b. Site Data Review and Preparation. 14
c. Model Building Process. 15
d. Scoring Process. 16
e. Model Ensembles. 17
f. Record Selection for Audits. 17
g. Audit Preparation. 18

III. RESEARCH METHODOLOGY 19
A. ANALYSIS OVERVIEW. 19
B RESEARCH QUESTIONS. 20
1. Is There a Relationship Between Fraud and CNIs? 20
2. Are There More CNIs in Supervised Records Than in Records Selected Randomly? 20
3. Is the Current Modeling Process Creating Improved Models? 20
C. PROCESS IMPROVEMENTS. 21
1 Analyze and Simplify Model Evaluation. 21
2. Create a CNI Knowledge Base. 22
3. Provide Improved Model Ensemble Options. 23
4. Recommended CNI Modeling. 23

IV. PROCESS EVALUATION 25
A. RELATIONSHIP BETWEEN FRAUD AND CNIS. 25
B. SUPERVISED MODELS FIND CNIS. 27
C. THERE IS AN ADEQUATE LIBRARY OF MODELS. 30
D. DISCUSSION OF RESULTS. 34

V. PROCESS IMPROVEMENTS 37
A. PERFORMANCE MEASURE EVALUATION. 37
1. Current Scoring Function. 38
2. Possible Alternative Scoring Functions. 42
3. The “Best” Scoring Function. 44
4. Implementing the Scoring Function. 46
B. CNI KNOWLEDGE BASE DEVELOPMENT. 48
1. CNI Data. 48
2. Reasons for Data Inclusion/Exclusion. 49
a. Sites Selected. 49
b. Audit Methods and Records Retained. 51
c. Fields Retained. 52
3. CNI Knowledge Database Development. 53
C. IMPROVING RECORD SELECTION. 54
1. Model Ensemble Discussion. 54
2. The Best Individual Classifiers. 55
3. Optimizing Ensemble CNI Detection. 56
4. The Underlying Classification Issue. 59
5. Sequential Selection Method. 61
D. SUGGESTED CNI MODELING PROCESS. 63
1. Differences between Fraud and CNI Data. 63
2. Train/Test Methodology. 64
3. Scoring Process and Ensemble Building. 66

VI. CONCLUSIONS AND RECOMMENDATIONS 67
A. CONCLUSIONS. 67
1. Fraud and CNIs Are Related. 67
2. Supervised Is Better Than Random Selection. . 67
3. The Current Modeling Process Has Reached its Limit. 68
B. RECOMMENDATIONS. 68
1. An Improved Scoring Function. 68
2. The CNI Knowledge Base. 69
3. Constructing Ensembles from Current Library. 70
4. Model Development Using CNI Knowledge Base. . 70
C. CLOSING REMARKS. 71

APPENDIX A. S-PLUS CODE FOR CHAPTER IV ANALYSIS 73

APPENDIX B. CLEMENTINE IMPLEMENTATION OF MODEL DEVELOPMENT SCORE FUNCTION 79

APPENDIX C. CNI KNOWLEDGE BASE FIELDS 83

APPENDIX D. FOUR SITES' MODEL NAMES AND CNI CLASSIFICATION RATES 89

APPENDIX E. GAMS CODE FOR OPTIMIZED ENSEMBLE 91

LIST OF REFERENCES 103

INITIAL DISTRIBUTION LIST 107

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure 1. Record Selection Process Flowchart 13
Figure 2. Supervised Versus Random CNI Findings 28
Figure 3. Boxplot of Scores From Four Sites 34
Figure 4. Optimized Classification Rates by Ensemble Size 58
Figure 5. Best Ensemble Classification Rates 59
Figure 6. Clementine Screenshot of Test/Train Palette 65
Figure 7. Clementine Score Development Screenshot 79
Figure 8. “Payment Info” Supernode 80
Figure 9. “NumCases” Supernode 80
Figure 10. “NumCasesFound” Supernode 81
Figure 11. “Score Determination” Supernode 81
Figure 12. “Score” Node Screenshot 82
Figure 13. Example Score Output Table 82

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1. Supervised Versus Random Results for Four Sites .. 29
Table 2. Variable Definitions for Mantel-Haenszel Test 30
Table 3. Model Scores for Specific Sites in 2001 32
Table 4. Comparison of Actual and Proposed Model Scoring .. 38
Table 5. Contingency Table of Voting Results 38
Table 6. Contingency Tables Representing the Four Data Runs Used in Model Development Score 40
Table 7. Example of One Model's Performance at SAC 44
Table 8. CNI Code Breakdown 49
Table 9. Site Audits Since Data-mining Inception 50
Table 10. Optimized Ensemble CNI Classification Results 58
Table 11. Neural Network CNI Classification Results 61
Table 12. Sequential Screening Ensemble Results on SAI Data 63

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGEMENTS

The author would like to acknowledge the financial support of DFAS, IR Seaside, for allowing travel to the Dayton site audit in February 2002.

The author wants to thank the following members of DFAS IR Seaside for putting up with his incessant questioning of the modeling process: Dave Riney, Randy Faulkner, Margot Wolcott, and LTC Chris Drews.

The author would like to extend a special thanks to Prof. Gerald Brown for his outstanding help with the optimization problem, and to Prof. Sam Buttrey for his guidance and patience during the performance of this research.

The author would like to acknowledge Lorrie and Alexandra Jenkins for supporting him at home during his many hours of analysis, research and typing. Without their presence, he would have been subsisting on Snickers and Diet Coke.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

In an effort to combat fraud, the Defense Finance and Accounting Service (DFAS) uses data mining to analyze millions of vendor transactions each year. The long time required to go from an audit to investigation and prosecution precludes DFAS from using fraud as a performance measure of data-mining success. Instead, the conditions needing improvement (CNI) found during site audits have been used to gauge supervised data-mining model success.

No detailed analysis had been conducted to demonstrate the validity of using CNIs to measure performance or to show the data-mining success with CNIs. The validity of using this method is verified by a thorough literature review which demonstrates a clear link between fraud and CNIs. Then the CNI results from the four site audits conducted during 2001 are analyzed to prove that supervised modeling is finding CNIs at a higher rate than random record selection. Given the demonstrated data-mining success with CNIs, the next phase of the research evaluates whether the current modeling process is producing improved models. By evaluating recent model scores, it is shown that models currently being constructed are not improving over those models produced for previous site audits.

With the modeling process having reached a plateau, four process improvement areas are explored. The first improvement is a reformation of the current model scoring system. Changes to both the model development score and the post-audit score are recommended. Additionally, the

model development score calculation is transferred to a Clementine stream to permit immediate calculation of model scores. The next process improvement is the development of an audit results (CNI) knowledge base, which can be used to evaluate current model performance as well as generate new models. The third process improvement entails a fresh look at the model ensemble process including an optimized model ensemble, second-tier classification methods and sequential record screening. The final process improvement is a proposed CNI modeling process that compares and contrasts CNI modeling to the current fraud modeling.

The goal of this thesis was to determine the success of data mining and then investigate potential improvements. We have shown that data mining is improving the audit process. As for improvements, some are presently being implemented, such as the Clementine scoring stream and CNI knowledge base, while others will require further investigation to be fully realized. The research presented by this thesis will improve an already successful program such that the data-mining efforts will further reduce taxpayer losses through fraud or misappropriation of funds.

I. INTRODUCTION

A. PURPOSE.

The purpose of this thesis is to evaluate the supervised data-mining process currently employed by the Defense Finance and Accounting Service (DFAS) Internal Review (IR) Seaside. The current data-mining efforts have been in place since 1999 following implementation assisted by an outside contractor, Federal Data Corporation (FDC) [Ref. 1]. Since that time, a dozen vendor-pay site audits have been conducted with the assistance of the data-mining team. Results from some of these audits have been collected and maintained in a Microsoft Access database. This useful data regarding the audited records may be regenerated into a knowledge base that can be used in model evaluation and for future modeling.

B. BACKGROUND.

The IR Seaside Office was originally formed as an independent investigative team known as Operation Mongoose. Mongoose was formed in an effort to stem the seeming flood of fraud cases that were occurring within the department of Defense and its disbursing activities. Oxendine [Ref. 2] conducted an analysis in 1999 of Department of Defense (DoD) fraudulent vendor payments and provided a good overview on the fraud problem that was occurring within DOD in the 1990’s.

In an attempt to leverage new data-mining technology, Operation Mongoose developed a data-mining project to search the vast numbers of DFAS vendor-pay transactions for potentially fraudulent payments. Sixteen known cases of

successfully prosecuted fraud were identified for use by the team. The records for these cases were collected and the associated 453 payments were compiled into a “knowledge base.” This knowledge base was analyzed for patterns and its records were classified into four fraud “types” that could then be used as a foundation for building supervised classification and prediction models.

After the inception of the data-mining project, the Mongoose team was integrated into the DFAS IR team. As part of IR, the team conducts data mining on vendor-pay transactions to provide focused record selection at DFAS site audits. The IR teams that visit DFAS sites then conduct a thorough audit of those records in search of Conditions Needing Improvement (CNI), overpayments, duplicate payments or possible fraudulent behavior. IR Seaside attends the audits to instruct the auditors about the data-mining techniques and to collect the audit results for future analysis.

C. RESEARCH QUESTIONS.

1. Primary Question:

Is the supervised modeling process improving the site audits conducted by DFAS IR?

2. Subsidiary Questions:

a. Is there a relationship between fraud and CNIs?

b. Are the records chosen by the data-mining process more likely to have CNIs than randomly selected records?

c. Is the existing set of models adequate to build a library of known successful models?

d. How can model selection be improved by including feedback from the audit results?

e. Are there alternative methods available for selecting supervised records?

f. Can the audited records database be used as a knowledge base for future modeling?

D. SCOPE OF THESIS.

The scope of this thesis will include:

1. A review of recent site audit results with the main focus on the four audits conducted during 2001.

2. A review of supervised model performance and comparison of audit results on records chosen by supervised models versus records chosen randomly.

3. Analysis of supervised model selection with an attempt to feed audit results back into the model selection and development process.

4. Analysis of methods to implement a model ensemble using the library of supervised models already built.

The scope of this thesis will not include:

1. The development of new models for DFAS IR.

2. A detailed analysis of the known fraud knowledge base currently being used to develop supervised models.

3. An analysis of other record selection processes currently used at DFAS IR.

4. Construction of models using the CNI knowledge base.

E. METHODOLOGY.

The methodology used in this study consists of the following steps:

1. Conduct a search of literature covering audit processes, fraud detection, data mining, and classification methods. This literature review will include journals, Internet resources, databases, and other library resources.

2 During experience tour at DFAS IR Seaside, evaluate the supervised modeling process from initial receipt of site data through the completion of record selection for auditing.

3. Attend a site audit to gain insight into the auditing process and to learn how the data-mining efforts might be focused to improve audits.

4. Collect and analyze data from previous audits.

5. Collect and analyze supervised models.

6. Evaluate the supervised modeling process.

F. ORGANIZATION OF STUDY.

This thesis is broken down into five chapters following the Introduction. Chapter II introduces DFAS IR and the techniques used to detect misappropriations and fraud. Following this overview, a detailed description of the supervised data-mining process is provided. This discussion steps through the modeling process from the initial phase of data receipt through the selection of records for auditing purposes.

The first part of Chapter III presents an overview of the issues examined in this thesis. Following these issues are the assumptions and hypotheses that this thesis will attempt to answer. Finally, the last section introduces proposals for an improved performance measure, more effective use of the model library and the creation of a CNI knowledge base. Chapter IV begins the analytical phase of the thesis. In this chapter, the three hypotheses will be addressed through literature review and statistical techniques. The literature review establishes a relationship between fraud and CNIs. The statistical analyses answer the questions of supervised modeling success versus random selection and the status of continued modeling on the fraud knowledge base.

Chapter V is a fresh look at some questions currently being explored by the data-mining team. It starts with an evaluation of the current model scoring function with recommended changes and a new software implementation. The next section describes the compilation of a new knowledge base of CNI data and possible uses for the data. Following that is an analysis of the model ensemble process in which an optimized ensemble is evaluated along with several novel ways to utilize ensemble results. The chapter concludes with a proposed modeling methodology for the CNI knowledge base. Finally, Chapter VI summarizes the conclusions and recommendations for process improvements. In addition, there is discussion on future research ideas associated with IR Seaside’s data-mining efforts.

G. EXPECTED BENEFITS OF THIS THESIS.

The initial benefit of this thesis is that it will demonstrate the success of the supervised modeling process. Additionally, by providing an analysis of the model library, ensembles are presented that might be used for future site audits and reduce the time spent on building new models. To aid in modeling, improved model performance measures are developed for incorporation directly into the modeling process. Finally, by creating a database of audit results, the modeling team will have a growing base of knowledge on which to improve the record selection process. Ultimately, this thesis attempts to improve the audit process of DFAS IR and thereby reduce the cost to DOD in overpayments and fraud.

II. BACKGROUND

A. BACKGROUND INFORMATION.

1. Description of DFAS IR Seaside.

DFAS is the agency that pays most DoD bills from contracting, travel payments, foreign sales and payroll disbursements. DFAS is one of the largest accounting agencies in the world, disbursing nearly one billion dollars every business day. DFAS was formed in January 1991 to eliminate redundant disbursement activities within the Defense Department. Prior to DFAS's inception, there were 338 accounting and finance offices worldwide. This excessive number of redundant systems and personnel cost the government 3.1 billion dollars per year in fixed overhead. In addition to this excess overhead, the large bureaucracy and the lack of standardization left the Defense Department vulnerable to fraud. The system has subsequently been reduced to 26 sites worldwide with offices in the United States, Japan and Europe. [Ref. 3]

During the early and mid 1990's there were a number of fraud cases, discovered mostly by accident, that pointed to systematic problems in the DoD payment system [Ref. 2]. This problem has been continually addressed since that time with improved internal controls, operational audits and system standardization. However, more proactive techniques were needed to show the American people that DoD was actively fighting fraudulent activity. In 1994 Congress created a new unit called Operation Mongoose whose sole purpose was to develop methods to detect and prevent fraud [Ref 2]. After several reorganizations, Operation Mongoose

is now the Seaside branch of DFAS IR, but its focus has not changed. Its number one priority is the discovery of fraudulent or problem payments within DFAS to save the taxpayers' money.

Since its debut, DFAS IR Seaside has developed analytical methods to find problem payments by exploiting the vast amount of payment information collected continuously by DFAS. Its agents work closely with the Defense Manpower and Data Center (DMDC) agency to gather pertinent data for analysis. IR Seaside assists the audit process with data analysis by searching for problem transactions such as duplicate payments, overpayments and fraud. The synergy developed by tying together these multi-agency functions has resulted in millions of dollars in duplicate payments being recovered, the initiation of fraudulent payment investigations, and the improved ability of auditors to identify conditions needing improvement at DFAS payment centers. [Ref. 4]

2. Literature Review.

Fraudulent payments are a problem not only within DoD, but also in practically all private and public institutions. A review of recent auditing literature finds numerous instances of fraud. Even in the daily news, one of the biggest stories of 2002 has been the collapse of Enron Corporation and the irregularities inherent in its auditing system. The overarching question is how to identify and prevent fraud by using the analytical tools available to the auditor.

Looking outside of DoD and into the private sector, detection of fraud is one of the primary responsibilities

of the professional auditor. The Statement of Auditing Standards (SAS) Number 82 is titled *Consideration of Fraud in a Financial Statement Audit.* SAS 82 details the responsibility of auditors to take action when they detect potentially fraudulent behavior during the conduct of an audit [Ref. 5]. To assist them in the auditing process, auditors are also required to use analytical procedures (APs) in the preparation, conduct and post-review of audits as outlined in SAS 8 [Ref. 6] and SAS 56 [Ref. 7]. Combining the various SAS requirements, it seems only natural that auditors would develop APs to assist in fraud detection.

Have private auditors had any success with their use of APs in detecting problem payments? Wheeler and Pany [Ref. 8] point out that studies using an *ex post facto* approach show that APs find numerous problems and should be applied more frequently. Busta and Weiberg [Ref. 9] demonstrate improved fraud auditing effectiveness using APs, specifically concluding that neural networks can be a valuable aid in the auditor's AP toolkit. Calderon and Green [Ref. 10] conduct a detailed review of accounting literature and show that APs used in fraud determination account for 16 to 40 percent of all findings in actual audit results. Apparently, private auditors are having success when using APs.

The use of APs to detect fraud is also an issue being addressed within other government agencies. The United States General Accounting Office (GAO) conducted a study of the techniques used by government agencies to detect or prevent fraud and improper payments. The study cites a

number of activities that are using data mining to detect abnormalities. For instance, the Illinois Department of Public Aid applies data-mining techniques to detect fraudulent billing and kickback schemes. Another case cited reveals how the Texas Health and Human Services Commission is using neural networks to identify fraudulent claims. The Texas commission successfully identified over six million dollars for recovery in fiscal year 2000. The GAO also reports on a number of other institutions and the data-mining techniques that are currently being used in fraud detection efforts. [Ref. 11]

With the recent focus on information technology, many agencies and corporations have spent millions of dollars to construct databases and implement data-mining activities. Is the effort really worth the cost? According to another GAO report, twelve government agencies required to report improper payments had estimated over sixteen billion dollars in improper payments as a result of operations during fiscal year 1999. Comparing the cost of millions of dollars to conduct data-mining efforts against the loss of billions of dollars makes the effort seem like a reasonable investment if even a small proportion of those lost dollars can be recovered. [Ref. 12]

B. DESCRIPTION OF IR SEASIDE'S DATA MINING.

1. Overview of IR Seaside Analytical Procedures.

IR Seaside currently conducts several different analytical techniques to identify problem payments. Before each site audit, the preceding eighteen months of site data is compiled for analysis. The IR audit coordinator decides how many records each selection method will screen for

detailed auditing with a typical breakdown including 30% duplicate payments, 30% supervised records, 10% unsupervised records, 20% related records, and 10% random records. Below is a brief description to enlighten the reader on the techniques used to identify records for further review.

a. Duplicate Payments.

Duplicate payments are payments made to a contractor, under a valid contract, that have already been paid at least once before. To identify duplicate payments the IR Seaside team evaluates all payments made at a DFAS site. This technique is initially computer-intensive in the comparison of all records with specific matching rules developed by the IR auditors. Site records are pairwise compared and several new record fields are generated. The new fields indicate whether a record shares common traits with another record in the database. If two records are nearly identical then they are flagged as potential duplicates. The second phase of the process entails an IR review of the flagged records. The Seaside auditors apply their experience to determine which records deserve attention during the upcoming site visit. Duplicate payments have been the most productive and visible aspect of the IR data-mining work with over $75 million dollars recovered to date [Ref. 4].

b. Supervised Modeling.

This technique is covered in more detail throughout the thesis. Described simply, the data-miners use a knowledge base of fraud to build predictive models. The underlying premise is that patterns in the fraud

knowledge base can be identified and exploited to predict potential fraud in the site data.

c. Unsupervised Modeling.

Unsupervised modeling is a comprehensive heading for all other modeling techniques used at DFAS that are not supervised. Some techniques used to date include clustering and pseudo-supervised/clustering. Additionally, some unsupervised records are chosen using subsets of the supervised modeling ensembles. This area of modeling is in its infancy at DFAS IR, is continually undergoing changes with each new site audit and could benefit significantly from further analysis.

d. Related Payments.

Related payments are records that are “related” to the records selected by the supervised models. When the supervised modeling process selects a record, a DMDC query then finds all other records related to the suspect record in the fields of payee, contract, address, or EFT number. All the related records are documented and the information is brought to the site audit. The related records may or may not be reviewed during the site audit depending on whether the audit reveals problems with its associated supervised record or if the data-mining team deems the record to be interesting.

e. Random Records.

Random record selection has been the traditional means of choosing records for DFAS audits. IR Seaside accomplishes random selection by assigning each record a random number from one to the number of records. The records are then sorted by this random number and the top

records are selected until the desired number of records is obtained.

2. In-depth Review of Supervised Modeling.

In order to understand the implications of the analyses conducted in this thesis, it is important for the reader to understand the supervised modeling process. Figure 1 outlines the key stages from initial site data reporting until the audit list of records is presented to the site. The remainder of Chapter II outlines the process from the development of the underlying knowledge base through the selection of records for an audit.

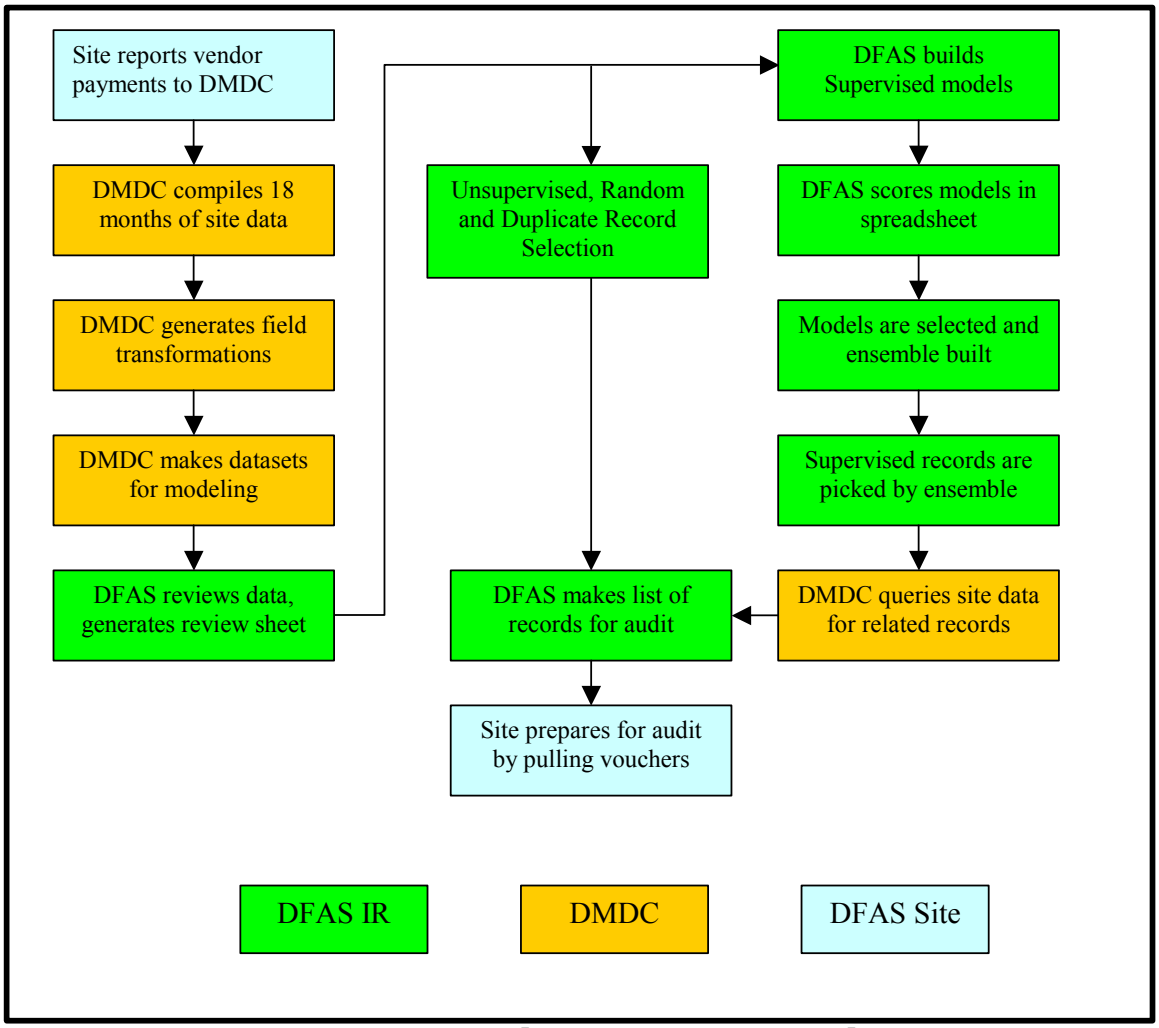


Figure 1. Record Selection Process Flowchart

a. Description of Fraud Knowledge Base.

The ultimate source of fraud knowledge used to generate supervised models comes from a knowledge base of prosecuted fraudulent activity. This knowledge base consists of 453 transactions from 16 known fraud cases. At the beginning of the data-mining program, these cases were collected from DMDC or from the actual transaction records. The data was analyzed using principal component analysis along with clustering techniques to group the payments for easier classification. [Ref. 1] This resulted in the fraudulent payments being broken into four fraud "types" for modeling purposes. These types were labeled as Big Systematic (BigSys), Small Systematic (SmallSys), Opportunistic (Ops), and Piggyback (Piggy). Because there were only 453 fraudulent payments, the analysts wanted the data to be utilized to its fullest extent. Therefore, the data was partitioned into eleven "splits" for model development in a way similar to a cross-validation scheme. Each split contains all 16 cases divided into three subsets in such a way that each subset contains transactions from each fraud type. These subsets are used for training, test and validation phases of model development explained later.

b. Site Data Review and Preparation.

As with any application, it is important to verify data integrity before analysis. Prior to modeling on any site data, the statistical information for each field is compiled and reviewed by the senior data-miner. That review includes looking for interesting traits such as negative date comparisons, missing data, implausible

entries and other abnormalities. The head data-miner then releases a comprehensive spreadsheet with recommendations as to which fields to avoid or to use in model development. Serious data integrity issues are reported to DMDC so the responsible site can be informed of the data entry problems that need to be addressed at the site.

In preparing the data, DMDC generates three subsets of data for each split: training, test and validation. The training set is used to construct models, which are then tested using the test set and evaluated using the validation set. Each of the eleven splits of fraud data is combined with a random sample of site records to create eleven test sets and eleven training sets. Each training set contains 4000 presumed nonfraud records from the site plus fraud records from the split. The test sets are similar, with 2000 site records and attached fraud split data. Finally, one set of nonfraud data for the validation set, consisting of ten percent of the site data, is used for all models. DMDC returns a Common Data Format (CDF) file that contains all the sets of data for modeling. This file is composed of eleven test sets, eleven training sets, eleven sets of fraud data for validation and one set of nonfraud data for validation.

c. Model Building Process.

After the Access database is delivered for model construction, the workload is distributed among the data-miners. Each split is assigned to an individual who will build a number of models to predict potential fraud in the site data. The splits are distributed differently for each site to preclude modelers becoming too split-specific in

their model development. Therefore, each modeler analyzes all the fraud data over several audits and a more robust model library is developed.

To build classification networks, the data-miners use the software package Clementine 6.5. Clementine offers a number of classification methods including classification and regression trees (with the C5.0 and C&RT algorithms), linear regression, logistic regression, and neural networks. The data-mining efforts have primarily used classification trees for their ease of understanding and neural networks for their ease of use. [Ref. 13]

The next step in the model-building process is model development. The miners build classification trees and neural nets using the splits assigned to them. The miners apply their audit experience from previous site data-mining efforts to develop what they consider the best models. Each modeler will build at least three models per split, and then enter the test and validation results into an Excel spreadsheet for scoring.

d. Scoring Process.

When the data-mining project began, the analysts lacked feedback regarding the effectiveness of their models. Therefore, they designed an *ad hoc* scoring function that has been in use ever since. The scoring process is briefly explained here with a more detailed analysis included in Chapter V. [Ref. 14]

After building a satisfactory model on the training data, the modelers run the test and validation datasets through the model. The result is contingency tables with counts distributed by rows of known fraud

status and columns of predicted fraud status for the test and validation sets. One major assumption is made when applying the known fraud label: that none of the sampled site data is fraudulent. Given the large number of payments and the belief that most payments are not fraudulent, this assumption is reasonable. The Clementine contingency tables are then manually transferred to an Excel spreadsheet. The model scores are calculated by Excel and consist of a weighted nonlinear utility function discussed in detail in Chapter V. All the model results are posted to a single spreadsheet to allow model comparison. The scores are then used as an objective factor in the subjective selection of models for the site's supervised ensemble.

e. Model Ensembles.

The next phase is to have all the models classify all the records in the site database. Models are selected based on the objective score along with the intent to evenly distribute splits, modelers and classification methods. The model team constructs a model ensemble made up of an odd number of models (recent sites used 11 models). The models are then built into a Clementine final voting stream and records are selected using a simple majority-voting scheme similar to that used in bagging [Ref. 15].

f. Record Selection for Audits.

With the modeling process complete, the entire eighteen-month database of site records is run through the model ensemble. Each model classifies each record and the predicted fraud classifications for each record are

counted. A true simple majority-voting scheme would classify all records that receive a majority vote as potentially fraudulent and worth review. However, audit team resources and time are limited, so only a fixed number of records can be selected. The team knows beforehand how many records can be reviewed, so they evaluate the ensemble results and chose a vote cutoff that returns approximately this many records. At the Dayton 2002 audit, eight out of eleven models was the cutoff due to an audit limit of approximately 140 supervised records. One problem with this selection method is that as the voting cutoff increases, voting blocs may become more dominant. Possible alternative methods are discussed in Chapter V.

g. Audit Preparation.

Records selected by the different techniques are referred to as candidates. The data-mining team sends the candidate list back to DMDC. DMDC then prepares and returns a list of the candidates and any related records to DFAS IR. Approximately two weeks prior to the site visit, DFAS IR forwards the candidate list to the audit site so that the records' documentation can be prepared for presentation upon the audit team's arrival.

III. RESEARCH METHODOLOGY

A. ANALYSIS OVERVIEW.

Data mining at DFAS IR has progressed to the point where the staff are efficient at data review, modeling and record selection. “Quick-look” subjective measures have been used to evaluate the success of their work, but no one has conducted a detailed analysis. This thesis is an attempt to objectively review the supervised data-mining efforts and to introduce possible process improvements

At first glance, one would think that a valid determination of the data-mining effectiveness would be the positive identification of fraud. However, because of the long time required between identifying potentially fraudulent records, investigation and prosecution, this performance measure is impractical. For this reason, the idea of using fraud detection as a performance measure of proactive fraud auditing is not seen as valid in the auditing literature. In his comprehensive book on proactive fraud auditing, Howard Davia resoundingly rejects fraud detection as a performance measure because of the historical difficulty of prosecuting fraud [Ref. 16]. He points out that proactive fraud auditing's greatest strength lies not in its ability to detect fraud, but more in its deterrent aspects. Unable to adequately measure success in detecting fraud, IR Seaside must develop another measure of effectiveness. This thesis’s research will show data mining’s effectiveness through review of prior audits, as well as develop more useful scoring measures and explore improvements to the current procedures.

B RESEARCH QUESTIONS.

1. Is There a Relationship Between Fraud and CNIs?

The measure used to evaluate the data-mining process to date has been the models' capability to identify CNIs during site audits. This seems appropriate at first glance, but it may not really address the underlying issue of finding fraud. The assumption has been made that a relationship exists between fraud and CNIs. To demonstrate this assumption statistically would encompass an entire thesis by itself. Therefore, to validate this assumption, we will not conduct statistical tests, but rather cite research that has demonstrated the relationship between record deficiencies and fraud.

2. Are There More CNIs in Supervised Records Than in Records Selected Randomly?

If the answer to question one above is correct then the supervised records should have a higher proportion of CNIs than the random records. Fortunately, the audits have included a random selection of records that have been reviewed. A comparison of the two methods' CNI findings will demonstrate the effectiveness of supervised modeling. In validating this assumption, a hypothesis test will be performed on the CNI/non-CNI findings for both supervised and random records at the four most recent site audits.

3. Is the Current Modeling Process Creating Improved Models?

Many hours and dollars have been spent in developing hundreds of models on a common knowledge base of fraud. Is it proper to assume that new patterns are still being identified in the data using the same classification methods? If that is not the case, then building more of

the same types of models on the same data may be pointless. A comparison of model scores will determine if the process is improving or stagnating.

C. PROCESS IMPROVEMENTS.

1 Analyze and Simplify Model Evaluation.

The current scoring process requires the manual transfer of approximately 100 values from Clementine contingency tables to an Excel spreadsheet. This process has a number of disadvantages. The first entails the design of the Excel spreadsheet and scoring function. Even though the DFAS modelers provided input, the contractors failed to document the function, which means that the modelers do not fully comprehend the spreadsheet's inner workings. Additionally, the manual transfer of data is time consuming and ripe for data entry errors which may result in miscalculated scores. Finally, because the scores are not readily available, miners lack immediate feedback on model performance when constructing models.

Following audit completion the team determines model success differently than during the model development phase. Each model is evaluated based on the percentage of records classified as potential fraud by the model that were subsequently identified as CNIs. This measure completely neglects to evaluate model performance on the records classified as potential non-fraud. This information is important and should be included in any performance measure.

It appears that there are two distinct scoring problems, but a closer look shows that a measure should be developed that can be used before and after audits. If

such a measure is applied then the post-audit model performance can be related, at least indirectly, to the model development score. It is desirable to determine successful model characteristics and integrate those traits into new models.

Finally, by understanding the current scoring process, it may be possible to calculate scores within Clementine. If the scores were calculated immediately while modeling then the data-miner would receive instant feedback. Additionally, having scores available in Clementine will reduce the problem of transcription errors because the final score information would be already compiled and ready for transfer to a spreadsheet.

2. Create a CNI Knowledge Base.

If all the patterns in the fraud knowledge base have been found using the current methods, then what is the next step? Adding new fraud cases could enhance the fraud knowledge base. IR Seaside is working with investigative services to obtain new fraud cases and will continue to pursue that effort. Another option, presuming that CNIs are related to fraud, is to develop the audit results into a knowledge base. Using CNIs will offer several advantages. The first is that the records in the current fraud knowledge base occurred before electronic transactions were popular, so the knowledge base is missing valuable transaction fields. In addition, after each audit, IR Seaside returns with that site’s audit results in their database. This means that the CNI knowledge base could be continually growing with new audit results, which include both CNI and non-CNI findings.

3. Provide Improved Model Ensemble Options.

With the process validated by proving the hypotheses above, the next step is to find ways to improve the ensemble process. With a growing knowledge base of CNIs, the potential exists to build ensembles based on their ability to detect CNIs, thereby providing a means of exploiting the hundreds of models in the library. The models used at the sites can now be evaluated on their real-world classification success, which can then be integrated back into building model ensembles. Several methodologies will be explored here including best model combinations, ensemble optimization, sequential screening and second-tier classification.

4. Recommended CNI Modeling.

With the potential for an ever-growing database of CNIs, the opportunity for a new modeling path exists. Presuming a link between fraud and CNIs, the modelers can build new prediction models to find CNIs. Although this method will have much in common with the fraud detection process, there are significant differences. The assumption of all site payments as nonfraud cannot be used with CNIs. Historically, there have been a significant number of CNIs at all sites. Additionally, the detailed analysis performed on the fraud knowledge base has not been performed on CNIs. This means that there may be undiscovered patterns within the CNI knowledge base that need to be further explored.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. PROCESS EVALUATION

A. RELATIONSHIP BETWEEN FRAUD AND CNIS.

The current process assumes the existence of a relationship between fraud and CNIs. The data-mining modelers use the fraud knowledge base to build fraud prediction models. However, the audit team does not search strictly for fraud. The primary focus of an audit is to find problems with the documentation, which are then recorded as CNIs and may be indicative of deeper problems such as fraud. The obvious question one must ask is whether there is a relationship between fraud and CNIs.

IR Seaside evaluates model performance on CNI identification because of the need to demonstrate to management how data-mining efforts are improving the auditing process. However, no objective analysis has shown that this relationship actually exists. Such an analysis would produce enough material to fill an entire thesis by itself. Consequently, we do not attempt a detailed analysis of that relationship. The accounting literature contains numerous articles that demonstrate the relationship between fraud and faulty documentation and that will be used as evidence to verify the base assumption of a fraud/CNI relationship.

Nita Crowder provides a quick overview of computer use for fraud detection in her article “Fraud Detection Techniques.” [Ref. 17] She discusses personal audit experiences and highlights techniques of data analysis that helped her discover fraud. Included in her analysis toolkit are simple techniques such as filtering data for

large purchases, comparing vendor addresses to employee addresses, and checking payments which are slightly below management approval level. She also discusses sophisticated techniques such as neural networks, expert systems and Benford's Law. Many of these techniques are also used at DFAS to find problems within payment documentation. The conclusion of her article points to a connection between documentation problems and fraud which helps validate our assumption.

In "Internal Fraud Leaves its Mark," Calderon cites research involving fraud detection. [Ref. 18] The article's most interesting point is the breakdown of how fraud cases were initially detected. The most common indicator of fraud was improper documentation, which was represented in 39% of the cases researched. In other words, the documentation had conditions not in accordance with procedures. These are precisely the conditions known as CNIs at DFAS. The author outlines the connection between fraud, internal controls and documentation. In an organization with strong internal controls there are fewer documentation errors. For our purposes, it can be concluded that when CNIs are observed, the internal controls have failed and the potential for fraud is increased. Once again, this article suggests an underlying relationship between CNIs and fraud.

The next article proceeds beyond the link between fraud and CNIs and explains how this relationship can be used in detailed analysis. In their article "Signaling Fraud by using Analytical Procedures," Calderon and Green explain how the relationship between fraud and

misstatements can be used in fraud detection. [Ref. 19] They cite previous studies and stress one example where 40% of all errors encountered by auditors were identified using APs. Additionally, they point out numerous studies where APs identified a significant portion of misstatements during audits.

In evaluating these articles, in addition to those presented in Chapter II, it becomes apparent that three assumptions are held in the auditing profession. The first is that documentation problems are associated with fraud in a significant number of the cases identified. The second is that finding the problem documentation will help auditors find or prevent fraud. The third is that APs can be used to detect documentation problems and thereby detect fraud. By applying the logic of these three assumptions, it is completely reasonable to evaluate the fraud model detection performance by comparing audit CNI results. Therefore, the assumption of a relationship between fraud and CNIs appears valid and it is now acceptable to proceed with the remaining hypotheses of this chapter.

B. SUPERVISED MODELS FIND CNIS.

Having shown a link between fraud and CNIs, the next step is to determine how well the model ensembles are finding CNIs. Fortunately, the data-mining team has included a random selection of records in its audits. Using the random and supervised records it is a simple matter to compare the two methods. If the ensembles are successfully finding CNIs, then one would expect that the percentage of records with CNIs in the supervised sample would be larger than in the random sample. Figure 2 shows

the comparison of CNI findings for supervised versus random records at four site audits conducted during 2001.

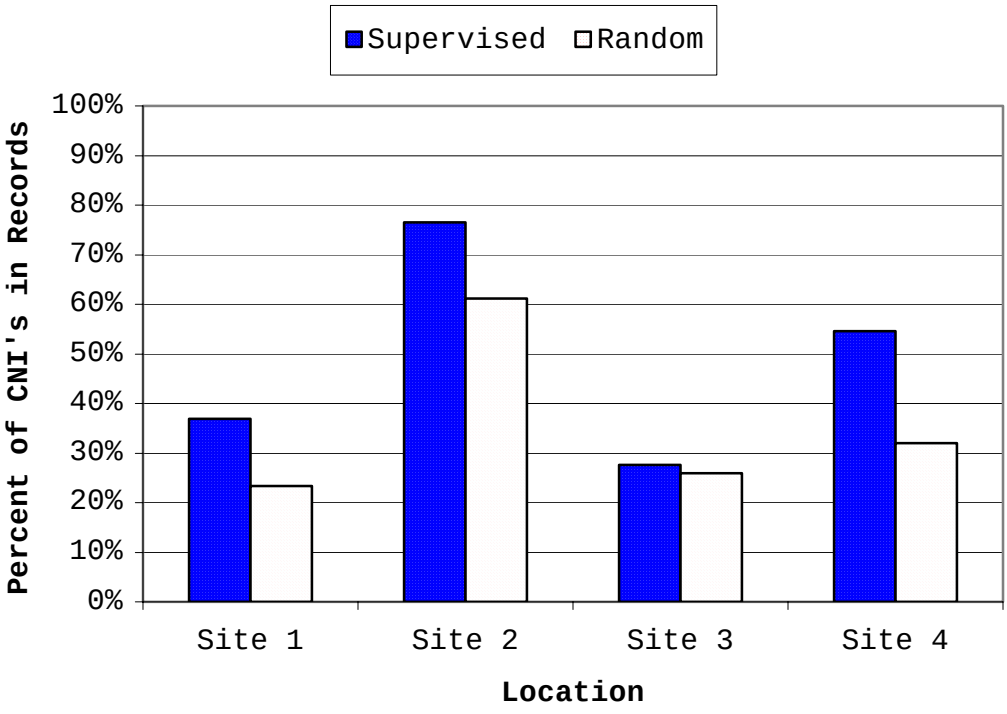


Figure 2. Supervised Versus Random CNI Findings

It is apparent that at each site the supervised records have a larger percentage of success at discovering CNIs. To verify this difference, a Mantel-Haenszel Test is used on the four sites' data. Table 1 is a breakdown of the data presented in Figure 2. The data in this table has fixed row margins because the numbers of supervised and random records is established prior to an audit. However, the column margins are random because CNI assignments are unknown when the records are selected. Therefore, as discussed by Conover [Ref. 20], the test statistic applied here will not include a continuity correction.

	Site 1		Site 2		Site 3		Site 4	
	CNI	Non-CNI	CNI	Non-CNI	CNI	Non-CNI	CNI	Non-CNI
Supervised	69	118	169	52	16	42	53	44
Random	7	23	63	40	13	37	16	34

Table 1. Supervised Versus Random Results for Four Sites

The null and alternative hypotheses are:

Ho: $p_{1i} \leq p_{2i}$ for all sites i

Ha: $p_{1i} \geq p_{2i}$ for all i and $p_{1i} > p_{2i}$ for some i

where:

p_{1i} – probability record at site i has a CNI when chosen by supervised technique.

p_{2i} - probability record chosen at site i has a CNI when chosen by random sampling.

The alternative hypothesis for this test is upper-tailed because we presume that the supervised modeling is finding more CNIs per record than random selection. Therefore, the alternative hypothesis states that supervised selection is at least as good as random selection for each site, with supervised being better at least once. Although these site populations are finite in size, they are large enough to neglect the finite population correction. Each site's table is separated into the variables shown in Table 2 for use in the test statistic calculation. The subscript i represents the site, meaning there are four tables within Table 1 that are like Table 2.

	Col 1	Col 2	Row Margins
Row 1	x_i	r_i-x_i	r_i
Row 2	c_i-x_i	N_i-r_i-c_i+x_i	N_i-r_i
Column Margins	c_i	N_i-c_i	N_i

Table 2. Variable Definitions for Mantel-Haenszel Test

The test statistic equations are shown in Equations 1, 2 and 3:

[1] $$Z = \frac{\sum x_i - \sum \frac{r_i c_i}{N_i}}{\sqrt{\sum \frac{r_i c_i (N_i - r_i)(N_i - c_i)}{N_i^3}}}$$

[2] $$p_{1i} = \frac{x_i}{r_i}$$

[3] $$p_{2i} = \frac{c_i - x_i}{N_i - r_i}$$

The calculation details for the Mantel-Haenszel test are found in Appendix A. The test statistic calculation results in Z=3.82, which results in a p-value=0.000067 when compared to a standard normal distribution. This provides strong evidence that at these four sites the supervised technique would find at least as many, or more, CNIs than the random selection process with at least one site finding significantly more.

C. THERE IS AN ADEQUATE LIBRARY OF MODELS.

The data-mining team has assembled hundreds of models since program inception, applying the same techniques and algorithms on the fraud knowledge base with minor changes for each site. The only thing that significantly changes

with each new site audit is the use of that site's data as nonfraud. When developing the models to detect fraud, the fraud input does not change. More than 300 models have been developed, with 52 unique models actually being used at the four sites analyzed in this thesis.

Given that there is a large model library available, is the modeling effort efficiently focusing the data-miners' work? Assuming that each modeler builds three to five new models for each site, then there are 20 to 30 new models created every audit. The creation of a base model takes about 10 to 20 hours for a modeler. The modeler's remaining models are typically spin-offs of the base model. The result is that each modeler spends about 30 to 40 hours developing new models for each site. Some quick arithmetic with rough numbers for one site shows:

Hours per modeler – 40 hours/site

Modeler salary per hour – 50 dollars/hour

Number of modelers – 6

Cost per site = $50*40*6 = $12,000

Once again these are rough numbers, but they make the point that approximately $12,000 is being spent to develop new models for each audit. This is a rather expensive proposition given that models are being built with roughly the same fields and using the same techniques as the models in the library. If the models are getting better, then one could argue that this cost is justified. However, if models are improving marginally, or not at all, then the efforts of the data-miners may be better utilized in other activities. In that case, the current library of models

could be harnessed to generate prebuilt ensemble streams. These prebuilt model ensembles would not require new model development, while their use may result in practically no loss in audit effectiveness.

Determining whether the models are improving is a difficult task. Many approaches can be taken given the assumptions that are made. To check for improvement here, the following criteria are used to develop a test strategy.

- Only models built specifically for a site are representative of that site's modeling effort.
- Model development scores that are currently used to rate models are an accurate indication of a model's worth.

The model scores for each site are shown in Table 3. Scores are on a scale from 0-600 and are explained in greater detail in Chapter V. There were seven models built specifically for Oakland, while the remaining sites each had five unique models.

Oakland (OA)	San Diego (SD)	San Antonio CAPS (SAC)	San Antonio IAPS (SAI)
589.67	552.93	568.78	565.98
584.96	552.93	558.47	545.05
558.46	533.00	550.19	536.53
553.07	529.98	539.84	535.38
547.19	528.85	530.09	534.22
545.70			
530.63			

Table 3. Model Scores for Specific Sites in 2001

An analysis of variance (ANOVA) test is inappropriate because the ANOVA test assumptions are not met. The samples do not appear to come from normal distributions and they have different variances between sites. Unable to

meet two of the basic assumptions of an ANOVA, the next option explored is a rank test. The test chosen here is the Kruskal-Wallis test because it allows for comparing multiple 2x2 contingency tables, while assuming only that the different population distributions have the same shape. The hypothesis test is therefore [Ref. 20]:

Ho: All of the site score population distribution functions are identical

Ha: At least one of the populations tends to yield larger observations than at least one of the other populations

The description of the test statistic and equations are shown below.

[4] $$\chi^2 = \frac{12}{N(N+1)} \sum_{i=1}^{k} \frac{R_i^2}{n_i} - 3(N+1)$$

[5] $$R_i = \sum_{j=1}^{n_i} R(X_{ij})$$

where:

i = site {1,...,k}

j = model

$R(X_{ij})$ = the rank order of model j from site i

n_i = number of models at site i

N = total number of models

The detailed results of the Kruskal-Wallis test are located in Appendix A for the interested reader. The χ^2 statistic calculation resulted in χ^2=4.12. When compared to a chi-squared distribution with three degrees of freedom this results in a p-value=0.25. This implies that there is insufficient evidence to suggest that the model score

distributions are different between sites. This in turn suggests that the model scores are not generally improving with each new site. Figure 3 shows the score distributions in chronological order from earliest site examined to most recent. No evidence of an upward trend is apparent.

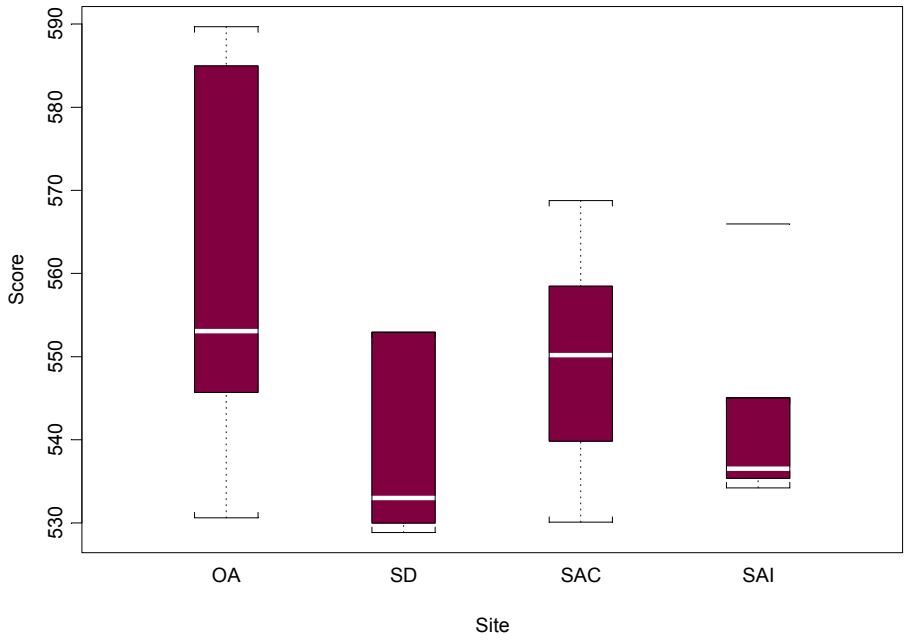


Figure 3. Boxplot of Scores From Four Sites

D. DISCUSSION OF RESULTS.

Here we summarize the analyses performed in this chapter. To start, a discussion of auditing literature showed a relationship between fraud and CNIs. This means that the current procedure of evaluating model success by IR Seaside is acceptable. Next, the results of previous audits were analyzed to prove that supervised data mining is improving the audit process. Records selected by supervised models have, on average, more CNIs than randomly selected records. This means that for future audits, DFAS IR would be better served if IR Seaside provided record

selection for all vendor-pay audits. Finally, the last analysis showed the supervised modeling process is not creating models that are improving with each new site audit. This suggests that the modelers have mined the fraud knowledge base to exhaustion using the current techniques. If this is the case, then the modeling process needs to proceed with different analysis techniques, add more fraud cases to the knowledge base, or use alternative means to model such as a CNI database. Ideas to improve the modeling process will be the focus of the remainder of this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

V. PROCESS IMPROVEMENTS

A. PERFORMANCE MEASURE EVALUATION.

Presently, IR Seaside uses an *ad hoc* scoring function in Excel developed during the data-mining program's inception. This spreadsheet implementation has been problematic in several ways. First, it requires manual transfer of approximately 100 numbers for each model built in Clementine. This data transfer is time-intensive and error-prone, requiring approximately ten man-hours per site. Additionally, scores are calculated when models are submitted for review and the modeling phase is essentially complete. Because the objective score is not readily available, modelers must go with gut instincts when building and choosing models. Finally, the initial model scores are not comparable to the performance measure used following an audit. There is no direct link to compare models from start to finish. All these areas require improvement to smooth model evaluation from initial development through audit completion.

This section attempts to evaluate the current scoring criteria used throughout a model's lifetime. The present scoring methods will be examined to determine their inherent strengths and weaknesses, while also exploring alternative methods. Table 4 briefly outlines the current scoring methods along with proposed methods for improvement.

	Current Method	Proposed Method
Development Phase	Spreadsheet score using weighted classification of test and validation dataset results.	Clementine stream using weighted classification of test and validation results.
Post-Audit Phase	Database score of CNIs found in records selected by the model.	Database score using weighted classification of all audit results

Table 4. Comparison of Actual and Proposed Model Scoring

1. Current Scoring Function.

All the data used in the scoring functions can be summarized in the 2x2 contingency table shown in Table 5.

		Predicted	
		Fraud	Not Fraud
Actual	Fraud	a	b
	Not Fraud	c	d

Table 5. Contingency Table of Voting Results

The model development scoring function uses the two ratios in Equations 6 and 7. The first is the sensitivity rate, which is the percentage of known fraudulent records classified as fraud. The second is the false alarm rate, which is the percentage of nonfraud records classified as fraud.

[6] $$Sensitivity = \frac{a}{a+b}$$

[7] $$False\ Alarm = \frac{c}{c+d}$$

Each test and validation run consists of payment and case results. This results in the creation of four contingency tables when evaluating a model during the development phase. When calculating the ratios by payment each transaction is treated as an individual entity. When

calculating the ratios by case, if any record of a given case is classified as fraud, then the entire case is identified as fraud. There are varying numbers of fraud cases (as discussed in Chapter 2) and nonfraud cases depending on the dataset. The four sets of ratios used to determine the score are shown in Table 6. The scoring function is presented below with the current weights shown in brackets. [Ref. 1]

Indices

i – Data type {p-payment, c-case}
j – Model run {t-test, v-validation}
k – Model parameter {s-sensitivity, f-false alarm}

Input Data

w_i – Case/Payment Weight {w_c = 0.3, w_p = 0.7}
w_j – Test/Validate Weight {w_t = w_v = 0.5}
w_k – Parameter Type Weight {w_s = 1, w_f = 5}
Gain - Scale Scores from 0-600 {Gain = 100}
Power – Exponential factor {Power = 2}

Model Parameters

S_{ij} – Sensitivity of data type i for model run j
F_{ij} – False Alarm Rate of data type i for model run j

Intermediate Variables

Score - Combined score used in rankings

Score Equations

[8] $$Score = Gain \sum_{i,j} w_i w_j (w_s S_{ij}^{Power} + w_f (1 - F_{ij})^{Power})$$

[9] $$S_{ij} = \frac{a_{ij}}{a_{ij} + b_{ij}}$$

[10] $$F_{ij} = \frac{c_{ij}}{c_{ij} + d_{ij}}$$

[11] $$\sum_i w_i = 1$$

[12] $$\sum_j w_j = 1$$

[13] $$\sum_k w_k = 6$$

			Payment (i = p)		Case (i = c)	
			Predicted		Predicted	
			Fraud	Nonfraud	Fraud	Nonfraud
Test (j = t)	Actual	Fraud	a_{pt}	b_{pt}	a_{ct}	b_{ct}
		Nonfraud	c_{pt}	d_{pt}	c_{ct}	d_{ct}
Validate (j = v)		Fraud	a_{pv}	b_{pv}	a_{cv}	b_{cv}
		Nonfraud	c_{pv}	d_{pv}	c_{cv}	d_{cv}

Table 6. Contingency Tables Representing the Four Data Runs Used in Model Development Score

A brief discussion of the scoring function is in order. When misclassification rates are low, the incremental changes per misclassification are larger due to the squaring of the sensitivity and false alarm rates. However, as more misclassifications are made, there is a smaller score decrease per misclassification. This forces modelers to work harder to improve model scores at higher values.

Another key point is the use of sensitivity and false alarm rates in the calculation. There are 2000 nonfraud payments compared to about 200 fraud payments in the test set. This disparity is more severe in the validation set where nonfraud payments include 10% of all site data. This means a difference between numbers of fraud and nonfraud payments of several orders of magnitude, which would overwhelm the fraud payments. Using the sensitivity and false alarm rates reduces the fraud and nonfraud classifications to the same scale.

The weights are another vital part of the score. The Test/Validate weights give equal weighting to the two data sets, which may lead to overfitting the data because the test set is iteratively used during model development. The Case/Payment weights value the payment ratios by more than twice the case ratios. This was done because the modelers felt that the cases should have an effect on the score, but that the cases were not as important as the individual payments. Finally, the parameter type weight gives false alarms five times more significance than the sensitivity. The original purpose for this breakdown was to limit the misclassification of nonfraud records to minimize excessive record selection and wasted time auditing records. The value of six as the weight sum appears to be arbitrarily selected by the contractors and presumably holds no significance.

The typical score for “good models” is in the 500 to 570 range. Models scoring lower are usually discarded and very few models score higher. The current parameter weights (w_k) significantly affect the scores because the false alarm weight is five times the sensitivity weight. If only one model were used to screen records, then it might be desirable to minimize the false alarm rate to reduce the number of records selected for audit review. However, the modelers use an ensemble and it will be shown later in the chapter why it is more important for models to identify the fraud while letting the ensemble remove nonfraud records.

2. Possible Alternative Scoring Functions.

There are alternatives to this scoring function. Is it more beneficial to completely replace the current function or just make adjustments? This is an important question, so the first step is to research literature on classification. Hand's work on classification rules [Ref. 21] and his combined work with Mannila and Smyth on data mining [Ref. 22] are excellent sources for initial investigation.

According to Hand, the most popular performance measure is error rate or misclassification rate [Ref. 21]. Simply put, error rate is the percentage of records misclassified in a dataset. However, one problem with simple error rate on a 2x2 contingency table is that it fails to address differences in the severity of the two misclassification types. In the accounting literature, these two types of misclassification are referred to as Type I and Type II error [Ref. 23]. Referring to Table 5, the error rates are calculated as follows:

[14] $$Type\ I = \frac{c}{a+b+c+d}$$

[15] $$Type\ II = \frac{b}{a+b+c+d}$$

[16] $$Error\ Rate = Type\ I + Type\ II$$

When using simple error rate, the question arises as to which type of error is more significant. If it were important to reduce the number of false positives that must be audited, then one would want to minimize Type I error. Conversely, if it were more important to identify all the

fraud, then one would want to minimize Type II error. At DFAS, the number of known fraud payments is substantially smaller than the number of nonfraud payments. Using simple error rates would allow nonfraud payments to overwhelm the fraud payments. In that instance, models that always classify every record as nonfraud would perform well because they would properly classify over 90% of the payments. However, because finding fraud is the purpose of the data-mining efforts, simple error rates seem inappropriate as a scoring function.

Another scoring option involves using only half of Table 5. This approach is currently used following site audits in which the rows of Table 5 change to known CNI status rather than known fraud status. As shown in Equation 17, models are evaluated based on the percentage of records that had CNIs out of those that it classified as potentially fraudulent.

[17] $$CNI\ Classification\ Rate = \frac{a}{a+c}$$

This method may seem appropriate, but it neglects all the records the model classified as nonfraud. Table 7 shows the results of a deceptive model from SAC. Using the CNI Classification Rate, the model looks good because it classifies 14 records as fraud and 12 of those had CNIs. This results in an excellent score of 85.7%. However, there were an additional 83 records with 41 CNIs missed by the model. The score is therefore missing a very important aspect of the model’s performance.

		Predicted	
		Fraud	Not Fraud
Actual	CNI	12	41
	Not CNI	2	42

Table 7. Example of One Model's Performance at SAC

Beyond the simple methods described above, there are more complex methods that use techniques such as logistic distributions, chi-squared tests, and response curves [Ref. 21]. However, a simple function is valuable here because it is important that modelers understand the score development and its interpretation. By using a simple scoring function, it will be easier for data miners to improve scores while developing models.

3. The "Best" Scoring Function.

Where does this discussion leave the current scoring function? It is relatively simple and the modelers understand its application, which is an advantage over more complex methods. The function accounts for all the contingency table data, which is an advantage over the post-audit score using CNI Classification Rate. The score includes weights to stress one type of misclassification over another, which is a step above simple error rate. Additionally, it avoids the fraud being overwhelmed by more numerous nonfraud payments, which would happen using a simple error rate.

The current function does have drawbacks, however. First, its spreadsheet implementation is time-consuming, requiring about ten man-hours to manually enter data, which equates to approximately $500 per site audit. To score a model, its Clementine data matrices must be manually

transferred from Clementine to the Excel spreadsheet. Typically the data is not properly ordered in Clementine so it must be rearranged before transfer. The manual data entry is ripe for mistakes that cause score errors. This could result in deleting a potentially excellent model or keeping a poor performer. Additionally, there may be several weeks' delay from model development to data transfer, thereby preventing modelers from receiving immediate feedback during model development.

The spreadsheet implementation is specifically designed for scoring models during the development phase. Because of this specialization, it is not used to rate and compare model performance on CNIs following audits. Such a comparison is desirable and would allow the data-mining team to find models that perform well before and after an audit.

On a mathematical level, the false alarm and sensitivity weights appear misplaced given the modeling team's desires and the ensemble approach of record selection. Presumably, the models should be finding as much fraud as possible while letting the ensemble process remove the nonfraud misclassifications. However, this is not occurring because the parameter weight for nonfraud is five times larger than the one for fraud. Most likely, models that found most of the fraud have been discarded just to reduce false alarm rates and thereby increase scores. For a single model, this is a reasonable approach because the auditors want to minimize time spent auditing false alarms. However, the ensemble approach to record selection screens out many records misclassified by

individual models. Therefore, it is important that individual models find the fraud so that the ensemble has an overwhelming majority vote for the fraud records.

Another issue is the artificial creation of nonfraud "cases." The site records are grouped by payee to assign the nonfraud case classification. The majority of nonfraud cases consist of one payment; typically 90% of the cases are composed of no more than two payments. Misclassifying some of these nonfraud cases causes only minor changes to the final score. On the other hand, the fraud data typically has only four cases in the test or validation set. If one fraud case is misclassified then the case sensitivity decreases by 25%, which results in a drop of 15 points on the overall score.

4. Implementing the Scoring Function.

The preceding discussion should make it readily apparent to the reader that there is room for improvement with the current scoring function. While it may not need to be replaced, implementation changes and weight adjustments are in order. The next section outlines a proposed scoring function that encompasses all phases of a model's use. Splitting the current scoring function into two separate functions will allow evaluating the fraud cases in the pre-audit scoring, while permitting the function to be used as a post-audit scoring tool. These two scores capture most of the same information thereby allowing a comparison, albeit indirect, from pre-audit to post-audit.

Outlined first is the proposed model development function. It still contains both the test and validation

runs, but with several differences. First, the false alarm rate based on the nonfraud case data is gone, which eliminates the artificiality of creating nonfraud cases. Instead, the test and validation runs now have a penalty that deducts points when fraud cases are missed. Another change to the case scoring is elimination of squaring the case sensitivity. If it were squared, missing the first case would penalize the model a whopping 44% of the total penalty. Therefore, the sensitivity is not squared; missing the first case is just as important as missing the last one.

The parameter weights now sum to one, thereby scaling the score from 0 to 100. This new scale seems more intuitive than the current 0-600. The weight recommendations are subjectively based on the desires of the data-mining team. The goal is to weigh fraud payments twice as heavily as nonfraud while including a minor penalty for missing fraud cases. The proposed scoring function is:

[18] $$Score = Gain \sum_j w_j (w_s S_{pj}^{Power} + w_f (1 - F_{pj})^{Power} - w_c (1 - S_{cj}))$$

[19] $$w_f + w_s + w_c = 1$$

[20] $$w_t + w_v = 1$$ {j = t for test, v for validation}

Recommended Weight Assignments

w_f=0.3 w_s=0.6 w_c=0.1 w_t=0.5 w_v=0.5 Power=2 Gain=100

Next is a post-audit scoring function that can be compared to the model development score. It is essentially the same scoring function, but with the case information removed. By using the same base function, it is possible

to compare model effectiveness from the development phase through the audit's completion.

[21] $$Post\ Audit\ Score = Gain(w_s S^{Power} + w_f (1-F)^{Power})$$

[22] $$w_f + w_s = 1$$

Recommended Weight Assignments

w_f=1/3 w_s=2/3 Power=2 Gain=100

The modified functions address the mathematical issues: the spreadsheet implementation problems are resolved by implementing the scoring function as a Clementine stream. Now that the function details are understood, building a stream for the modeler's palette is trivial. The scoring stream can then immediately score models as they are created. The details of the Clementine scoring stream are shown in Appendix B. As for the post-audit scores, individual model scores can be calculated in the DFAS Access audit database and immediately evaluated following an audit.

One benefit of a Clementine scoring stream is that the score output can be easily copied over to a single Excel worksheet. Each modeler can post his or her own model scores immediately and the head data-miner can track scores as the model development phase progresses. After audits, the post-audit model scores can be copied from Access into the same Excel worksheet for direct comparison within and between models. These changes to the scoring process will save time and simplify model evaluation.

B. CNI KNOWLEDGE BASE DEVELOPMENT.

1. CNI Data.

At each site, auditors complete a checklist in an Access database for each audited record. The checklist

includes preloaded categories for the auditors to describe any regulatory violations. The auditor can assign more than one type of CNI to a record simply by selecting all applicable categories. In the background the categories have two associated fields: CNI and comment codes. Each comment code is unique to the specific comment, but there are only four categories for CNI codes as shown in Table 8.

The records can be queried from the database either by individual transaction or by comment code. A query that extracts data by the comment codes may return a transaction more than once if it has more than one assigned comment code. A query that extracts the data by payment will return each transaction exactly once with the worst case CNI Code that it received. This means that even if a record has more than one comment, it will be listed only once in the table. This difference is significant and must be understood when building a CNI knowledge base.

CNI Type	**Code**
Significant Condition Needing Improvement (SCNI)	1
Condition Needing Improvement (CNI)	2
Observation (Obs)	3
No CNI (NCNI)	4

Table 8. CNI Code Breakdown

2. Reasons for Data Inclusion/Exclusion.

a. Sites Selected.

Table 9 lists the audits in which IR Seaside has been involved. It would be desirable to include all this data in a knowledge base; however, the data-mining process was still in flux at the earlier sites. Initially, the data-miners were actively training the auditors on the

Access database. Furthermore, CNI and comment codes were not standardized. Including the data from all the sites would require substantial data cleaning. Even if it were possible to adequately cleanse the data, there is no guarantee that the interpretation of the data was consistent at the early sites. Therefore, it is necessary to carefully select sites for inclusion while excluding sites that are more problematic.

DFAS Site	Date	Keep in database?
San Diego	Feb 2000	No CNI data
San Bernardino	Feb 2000	No CNI data
Denver	Feb 2000	No CNI data
Oakland	Apr 2000	No
Omaha	Jul 2000	No
Pensacola	Aug 2000	No
Dayton	Dec 2000	No
Oakland (OA)	Jan 2001	Yes
San Diego (SD)	Apr 2001	Yes
San Antonio IAPS (SAI)	Jul 2001	Yes
San Antonio CAPS (SAC)	Jul 2001	Yes
Dayton	Feb 2002	(Data Unavailable)

Table 9. Site Audits Since Data-mining Inception

The last three sites of 2001 were a good baseline with which to start developing a knowledge base. Issues that had been resolved by the time of the SD audit were field standardization, common model annotation, common field derivation and meaningful model naming. SAI and SAC were identical in all these aspects, but SD had some minor differences that were easy to fix. However, in a desire to maintain potentially useful data, OA was deemed acceptable. While it was at the cusp of the standardization efforts,

its data is not drastically different. Many OA field headings were different but contained the same information as the other sites. Additionally, some fields that were used at OA had subsequently been phased out. By overcoming these surmountable obstacles the OA data was included, thereby increasing the number of records by over 30%.

b. Audit Methods and Records Retained.

The question of which records should be included in the database is important because not all CNIs are created equally. The key is to include records that have received a thorough screening and an appropriate CNI classification. How a record is selected determines how thoroughly it is audited. The auditors consistently evaluate three record types at each site: those chosen with a supervised technique, those chosen at random and those chosen with an unsupervised technique. Auditors carefully scrutinize all information in these transactions while completing the Access checklist. Whether or not the record has a CNI, we are assured of an accurate classification based upon the auditors' findings. Therefore, these three record types should be included in a CNI knowledge base.

Records of the remaining two types typically do not receive the same attention to detail. The duplicate payment records are usually submitted to the site only for review as a duplicate payment. There is no guarantee that an auditor will have reviewed it in accordance with the checklist. Along the same line of reasoning, related records may not be audited unless a problem exists with a record's supervised parent. A related record classified as

non-CNI may or may not have been audited. However, the converse is not true: if a related record has a CNI then an auditor must have reviewed the record. Therefore, it would be unwise to include these two types for analysis because there is no guarantee that non-CNI records have been audited.

Taking these points into consideration, is it beneficial to include all the site records in the database? All five types of records are readily available. Rather than reject data that may prove valuable later, it is better to include all records from the sites. However, when using the data for a particular analysis it is important that the modelers understand the need to filter records that do not apply to that analysis.

c. Fields Retained.

Having decided which records to retain, the next step is to identify applicable fields. Not all sites had the same standardized data fields; therefore, site fields must be compared prior to building a knowledge base. Tallying the different field names results in 219 possible fields. Many fields in the OA database had the same information as fields in the SD, SAC and SAI databases, but with different names. Identifying the fields with the assistance of the head data-miner eliminated these inconsistencies.

OA also contained fields with no corresponding field at the other sites or which were no longer in use. Additionally, some of these fields contained all zeros or N/As and were therefore useless. These fields were deleted altogether. Appendix C contains a detailed table of which

fields were available at each site, which fields were kept and any significant comments related to the field status. The end result is the retention of 161 data fields with the addition of an “AUDIT_SITE” field that identifies the site from which a record came.

3. CNI Knowledge Database Development.

Having evaluated the records and fields from the four sites, the final step is to generate a single data table of audit records. When creating this table it must be flexible enough to allow introducing new site data following future audits. To facilitate expansion, the fields were structured around the San Antonio data because it has the format expected of future site audits.

The following methodology was employed to create the new knowledge base. First, for each site a query was created that extracted records on a per-payment basis. Each record has one entry with all the fields including a CNI field composed of the worst CNI Code associated with the record. For instance, if one record has two comments that correspond to CNI codes of 1 (SCNI) and 2 (CNI), the new record entry includes all the payment data only once with an overall CNI code of 1.

To build the CNI knowledge base, an empty table was generated in Access that was formatted to meet all the current field specifications. Then, append queries were conducted to include each site’s data in the table. With all four sites appended, the knowledge base is complete and ready for employment. Additionally, it will be a simple matter to update the current knowledge base following future audits. The head data-miner can simply generate the

same query for the new sites using the query already built into the IR Seaside audit database.

What benefits are there to having a CNI knowledge base? First, it can be employed to evaluate current models' abilities to predict CNIs at recent sites and thereby determine model effectiveness. Furthermore, the data-miners can use the CNI knowledge base to build new models for audit record selection. This is where the expansion of the knowledge base will come into play because it will continue to grow with each new site audit.

C. IMPROVING RECORD SELECTION.

1. Model Ensemble Discussion.

Record selection is currently achieved using a voting ensemble in which models are equally weighted. The individual models classify records as fraud (1) or nonfraud (0) and then the model votes are summed for each record. The result is an integer value between zero and the number of models (11 in SAC/SAI, 23 in SD/OA). Any records receiving a majority vote are classified as potentially fraudulent. For instance, at SD 1755 records received a simple majority of votes. However, resources (auditors, time, and money) are limited during an audit so not all 1755 records can be screened.

Before each audit, the data-mining team establishes a target for the number of supervised records that can be reviewed given the available resources. To reach this goal, the team sets a vote tally that becomes the cutoff for record review. For example, at Dayton in 2002 it planned to review 140 supervised records. After the model

votes were tallied, the appropriate cutoff to reach that goal was determined to be eight votes.

Although the number of records selected is kept to a manageable number, the elevated cutoff does come with a cost. For instance, Dayton records were selected if 8 of 11 models classified the record as fraud. With the cutoff at nearly 75% of models, the potential exists for voting blocs to take over the selection process. If 4 of 11 models vote together by consistently classifying records as nonfraud, they may dominate the voting and reduce the benefit of the voting scheme. The question then becomes how best to have the ensemble vote, while still controlling the number of records that are labeled as potentially fraudulent.

2. The Best Individual Classifiers.

With the introduction of a CNI knowledge base, there are now additional means of assessing model performance. Consider an ensemble built from the best individual classifiers of CNIs. This is similar to the way in which models have been evaluated following each site audit. However, with the CNI database, it is possible to assay model performance on more diverse data and then compare models to one another. To demonstrate this technique, we initially conducted a comparison of models using simple correct classification rates of those records that were selected by the supervised method. The results of this comparison are shown in Appendix D.

Knowing the capability of individual models may be interesting, but the best ensemble may not use all the models with the best classification rates. The ensemble's

purpose is to provide an output that enhances the models' CNI detection capability while compensating for their inability to screen non-CNIs. If an ensemble used only the best individual classifiers, it might be no better than the best individual model. The goal is to combine independent models to create a classification tool that is better than the sum of its parts [Ref. 1]. Therefore, although it may be useful to compare models individually, it is more important to create an ensemble such that overall classification improves.

3. Optimizing Ensemble CNI Detection.

How well can the simple majority-vote ensemble work? To answer this question, we propose a simple integer linear program to optimize the ensemble CNI classification rate. The linear program will choose an optimal subset of the 52 unique models employed at the four sites. To solve the linear program a software package called the General Algebraic Modeling System (GAMS) Interactive Development Environment is used [Ref. 24]. The detailed GAMS formulation is shown in Appendix E with a description and results provided here.

To implement the GAMS program, the supervised model records discussed in Section V.B were used as the voting data. This data consists of 563 records of which 308 were classified as CNIs. Each record had an actual CNI value that was 0 for non-CNIs and 1 for any other CNI code (cni_r). Each model voted on each record with a 1 representing a fraud classification and a 0 representing nonfraud ($vote_{r,m}$). Proper classification occurs when the vote majority agrees with the CNI rating. The program changes

ensemble size from 3 to 21 models by increments of two (enssize). This permits determination of the best ensemble as well as optimal ensemble size. The model formulation is shown below.

Indices

r	record
m	model

Data

$vote_{r,m}$	1 if fraud predicted, 0 otherwise
cni_r	1 if CNI found, 0 otherwise
enssize	number of models to select in optimal ensemble
majvotes	(enssize/2)+1 majority votes for ensemble size

Variables

$CLASSIFY_r$	correctly classify record r
$SELECT_m$	select model m for ensemble

Formulation

[23] $\max_{CLASSIFY,SELECT} \quad z = \sum_r CLASSIFY_r$

[24] subject to $\quad majvotes * CLASSIFY_r - \sum_{m|vote_{r,m}=cni_r} SELECT_m \le 0 \quad \forall r$

[25] $\quad \sum_m SELECT_m \le enssize$

[26] $\quad CLASSIFY_r \in \{0,1\} \quad \forall r$

[27] $\quad SELECT_m \in \{0,1\} \quad \forall m$

The number of possible model combinations in the case of eleven models is 52-choose-11 or 6.0×10^{10}. In evaluating the ensembles, classification rate is used due to its simplicity and ease of interpretation. Reported along with the correct classification rate is a contingency table similar to Table 5. The best ensemble classification performance attained was 398 out of 563 records. This results in a 70.7% classification rate and occurred with both 7-model and 11-model ensembles. The correct

classification rates decrease as models are added or removed from this optimum. Figure 4 shows the ensemble size versus correct classification rate and the contingency table for the best 11-model ensemble is shown in Table 10.

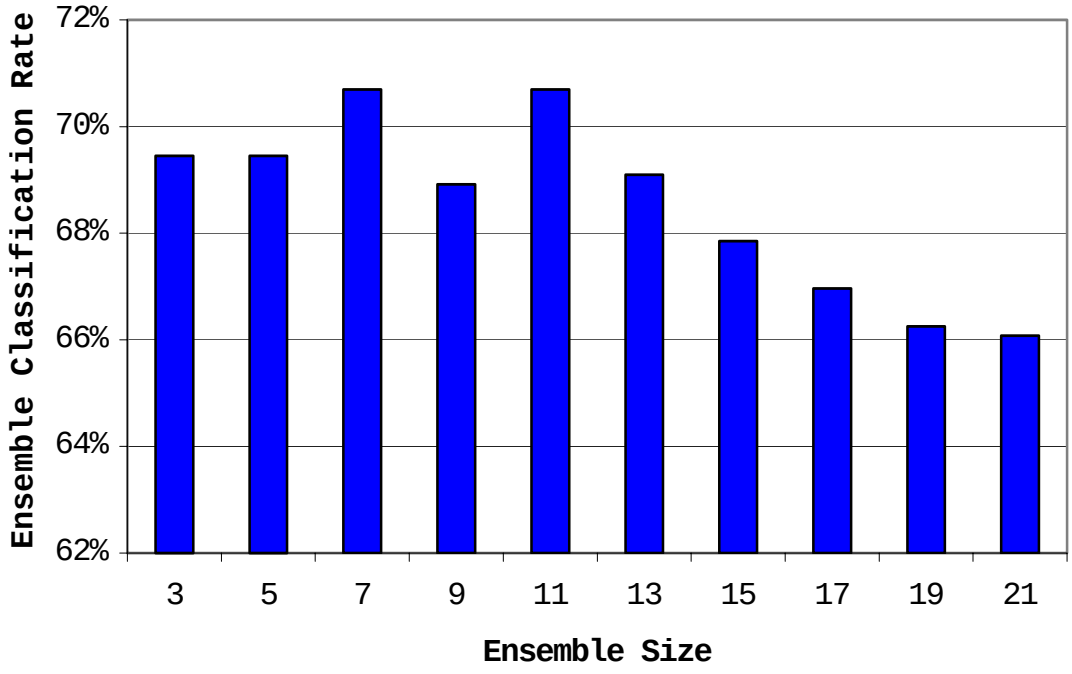


Figure 4. Optimized Classification Rates by Ensemble Size

		Predicted	
		fraud	nonfraud
Actual	CNI	216	92
	non-CNI	73	182

Table 10. Optimized Ensemble CNI Classification Results

Figure 5 shows the individual model classification rates with the ensemble classification rate represented by the bold line at 70.7%. This plot clearly shows that the ensemble is classifying better than any of its individual models. Furthermore, the models in the optimal ensemble

are not the best 11 models from Appendix D. Rather, the model blend appears to be balanced to create a better ensemble. However, we must caution the reader that this ensemble is presumably over-fit to this data and that its accuracy could decline when presented with unseen data. If this is the best model ensemble available then perhaps IR Seaside is doomed to classify CNIs with no better than 70% accuracy.

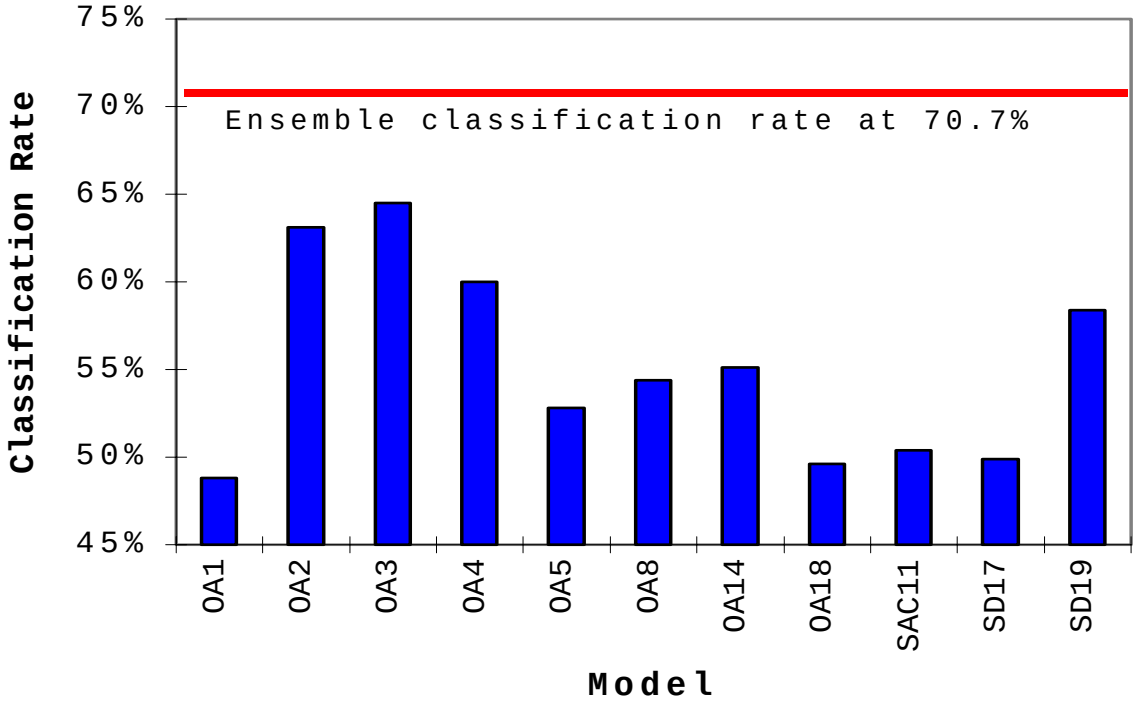


Figure 5. Best Ensemble Classification Rates

4. The Underlying Classification Issue.

If building and applying a model ensemble is an optimization problem, then a classification rate of 70.7% and the contingency table results in Table 10 seem unimpressive. Perhaps a majority-voting scheme may be incapable of identifying subtle patterns in the records or maybe there just is not enough information in the data. However, combining model votes might be viewed as yet

another layer in record classification if model votes were treated as additional predictor variables. Then we can build a “second-stage model” that combines model votes to produce a single fraud classification.

The idea investigated here is a second-stage regression neural network. Individual model votes are the inputs and the network has an output that ranges from 0 to 1. An output of 1 indicates that the record is worthy of review, while a 0 indicates that the record is clean and requires no further action. The regression network’s output is continuous and should fall somewhere between 0 and 1 due to the inputs being 0 or 1. By doing it this way, the team does not need to set a discrete number of votes as a cutoff for record selection. Instead, they would sort the output from highest to lowest and take the desired number of records from the sorted list.

In order to compare results, the models in the optimized ensemble from Section V.C.3 were used. The model outputs were the predictor variables to train the neural network. The response variable is the CNI values condensed in the same way as for the optimized ensemble. That is, any CNI is treated as a problem (1) while the non-CNIs were good payments (0). The statistical package used was S-Plus [Ref. 25] along with a dataset consisting of model votes and CNI codes for all four sites. The neural net was built using the neural network library of Venables and Ripley [Ref. 26]. The outputs produced by the network resulted in a correct classification rate of 92%. This result was determined by labeling any record with a net output >0.5 as

a CNI and any that had ≤0.5 as a non-CNI. Table 11 contains the results.

Comparing this outcome to the optimized ensemble using a simple majority vote results in an increase in correct classification rate of nearly 22%. Since each of the two techniques used all the data to classify the records, each method is presumably over-fit. However, building this neural net proves that records misclassified by the majority vote may still be identified via other classification methods.

		Net output >0.5 (Predicted CNI)	Net output <0.5 (Predicted non-CNI)
Actual	CNI	291	17
	non-CNI	28	227

Table 11. Neural Network CNI Classification Results

Opportunities exist to improve the record selection process. If using second-stage classifiers to combine model votes, then the process should be made robust by splitting the data into train and test sets or using some type of cross-validation method. Relevant techniques are outlined in, for example, Hand's Construction and Assessment of Classification Rules [Ref.21].

5. Sequential Selection Method.

Another technique selects just the records that received votes from all ensemble models. There were very few of these records at the four sites analyzed because of the large number of models in the ensembles. However, if we were more selective in choosing models, we might perform sequential model screening. All site records would be

submitted to the first model and all that are predicted to be fraud would be passed on to the second model. In the final Clementine stream, each model would see only the data that was classified as potential fraud by all preceding models. This process would continue until only the records that were predicted to be fraud by all ensemble models remained. Model selection would be based on the model's ability to find as many CNIs as possible. Revisiting Table 5, the goal is to find models with minimal b cell values in the contingency table. The consequence of this method is that most CNIs are retained after each pass while mostly non-CNIs are removed.

Examination of the SAI data shows that the number of records can be reduced from 132,699 records to 216 records by using just the three most CNI-sensitive models. The three models selected had sensitivities of 100%, 100% and 87.5% on the SAI supervised records. The site data contains the 56 audited records comprised of 14 CNIs and 42 non-CNIs. In reducing the records for review, as shown in Table 12, only 2 of the 14 CNIs are missed for an ensemble sensitivity of 87.5%. On the other hand, the non-CNIs are sequentially screened such that 29 of the 42 non-CNIs are rejected for an ensemble false alarm rate of 31.0%. Combining these results gives an overall correct classification rate of 73.2% for the audited records. Of the 216 selected records, the remaining 160 unaudited records have an unknown status, but this three-model ensemble points to the need for further review. This 73.2% classification rate shows that sequential selection is one more record selection alternative that is worthy of further exploration.

		Predicted	
		Fraud	Not Fraud
Actual	CNI	12	2
	Not CNI	13	29

Table 12. Sequential Screening Ensemble Results on SAI Data

D. SUGGESTED CNI MODELING PROCESS.

With the completion of the CNI knowledge base, the capability now exists for modeling on CNIs. IR Seaside has developed standard operating procedures for modeling on the fraud knowledge base, but some of the assumptions made and methods used may not apply when modeling with the CNI data. This section will address some of the more obvious issues.

1. Differences between Fraud and CNI Data.

As previously stated in Chapter 2, the fraud model development process assumes that sampled site data is not fraud. However, the assumption that site data has no CNIs would not be justified. From Figure 2 it is readily apparent that randomly selected records have greater than 20% CNI rates (CNI Codes 1-3) for all four sites. Obviously, it must be assumed that the site data has a significant proportion of CNIs. Instead of combining site data with knowledge base data, it would be more appropriate to develop models solely using the CNI knowledge base. Following model development, an ensemble would be used to select records for review from the entire site population.

Another difference between databases is the detailed statistical analysis already conducted on the fraud. The records have been evaluated and clustered into different fraud types that are used when building and scoring models. Conversely, the CNI knowledge base has not been subjected

to such scrutiny. However, these records naturally fall into the four CNI categories shown in Table 8. For the time being this might be useful, but a more detailed analysis of the CNI data may be in order to gain insights into relationships between CNI types and the different records and fields.

2. Train/Test Methodology.

Currently, three subsets of data are used in model development: training, test and validate datasets. Models are built using the training data, and then initially evaluated using the test data. At this point, model development is an iterative process; if the model performs poorly on the test data, the modeler will retrain the model on the training data. This process repeats until the modeler is satisfied. Consequently, the model error rates on the test data are over-optimistic because this data was used, albeit indirectly, to train the model. At the conclusion of this process, the model classifies the validation data to determine model effectiveness and then the validation and test results are used to score the model.

When developing CNI models this method can be applied by dividing data into training and test subsets. Instead of generating three sets on eleven unique splits, the CNI data could be randomly separated prior to each model development. By randomly generating these subsets, there is less chance of models being highly correlated. Clementine can automatically generate the training and test sets and store them in separate files when the data is imported. The modeler would then use a model training

stream to build the model. Following model development, the model would be placed into the test stream for evaluation and immediate calculation of an objective score. Figure 6 shows an example of a model development palette using the CNI knowledge base.

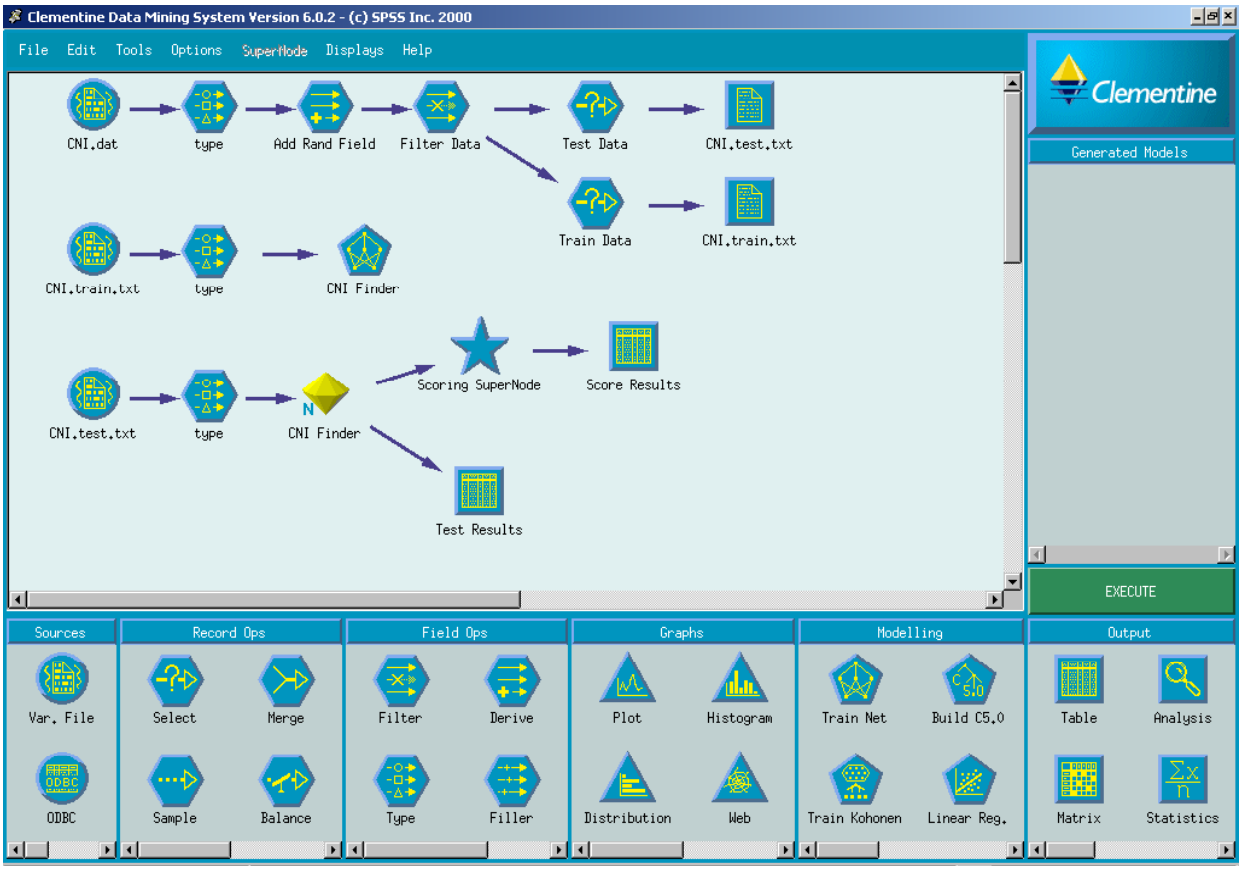


Figure 6. Clementine Screenshot of Test/Train Palette

A train, test and validate breakdown could help minimize problems transitioning to CNI modeling. However, depending on the subset of the CNI knowledge base used, the three development sets may each have to be small. For example, the current CNI knowledge base only has 563 supervised records. Equally distributing this data into training, test and validation subsets would result in only 189 records in each subset. This fairly small number seems

inadequate to begin building and evaluating models. However, as the database grows the potential for three distinct development data groups will also grow. For instance, by adding just three more audit sites to the CNI knowledge base, the supervised population should exceed 1000 records, thereby providing suitable subset sizes.

3. Scoring Process and Ensemble Building.

Evaluating the CNI models doesn’t pose a significant difference from the scoring proposed in Section V.A. If using only test and training datasets, the model’s test results could be used as in the proposed post-audit score in Equation 21. If using a train, test and validate approach, the models could be evaluated using a weighted score of the test and validate results much like the proposed model development score in Equation 18, only without the case penalty. In either instance, the functions already proposed for scoring would prove capable.

Ensembles may be built as they are now or by using one of the alternatives discussed in Section V.C. The use of an ensemble is a perfect fit for this modeling process because each model is built and evaluated on different subsets of the knowledge base. Using an ensemble would therefore capture the strengths of each model while reducing bias, variability, or both.

VI. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS.

1. Fraud and CNIs Are Related.

The auditing literature contains ample evidence demonstrating that a significant proportion of detected fraud has documentation discrepancies. This link between CNIs and fraud provides a launching point for IR Seaside to evaluate the success of their fraud detection models. Nevertheless, CNIs should not be the only tool used to evaluate model success, because the ultimate goal is to root out fraud. When data-mining efforts lead to the successful prosecution of fraud, then IR Seaside will have a direct means of measuring the success of their fraud detection efforts. In the meantime, CNI results are an appropriate measurement tool. Besides, even if the data mining does not lead directly to finding fraud, the project's deterrent aspect and the reduction in CNIs are valuable aids in preventing fraud.

2. Supervised Is Better Than Random Selection.

Selective record screening is proving to be a valuable aid to the DFAS IR audit teams, because the supervised modeling process is finding CNIs. This project successfully incorporates sophisticated technology to search the vast stores of DFAS site data. Rather than using outmoded, traditional random selection of a handful of records from among many thousands, supervised modeling targets those records that seem to require a detailed look. DFAS IR should exploit this record selection process all the time, not just at sites targeted by IR Seaside. If IR

Seaside creates predeveloped ensemble streams, the team should be able to generate audit lists in less than a week, rather than the many weeks the current process takes.

3. The Current Modeling Process Has Reached its Limit.

Evaluating model scores using the current performance measure shows that model development is not improving. A look at Figure 3 shows that model scores for year 2001 sites are not increasing with time. If the model scores are static then much effort could be saved by implementing novel strategies using the current library of models. Extensive effort and cost has gone into the production of this model library. There is an adequate set of models available to build canned ensemble streams which could then be rapidly deployed to support all DFAS IR audits.

B. RECOMMENDATIONS.

1. An Improved Scoring Function.

The current scoring function is inadequate and its present spreadsheet implementation is cumbersome and error-prone. The proposed alternative model development function will save time by using Clementine to rapidly calculate a score, thereby saving many hours of data transfer. Calculating the score in Clementine also gives immediate access to an objective score during development, which will enable modelers to improve the model while it is under construction. Additionally, by applying the same baseline function throughout the life of a model, the scores can be directly compared within and between models. This enhances the usefulness of the scores, thereby allowing feedback from the audits to improve the modeling process.

2. The CNI Knowledge Base.

Hoping to take advantage of audit results, the data-mining team felt that a CNI knowledge base would be valuable in several ways. First, the CNI data can be used to evaluate ensemble construction using the current models. This was demonstrated in Section V.C where an optimized ensemble was found and where alternatives to majority-vote ensembles were discussed. The CNI database adds flexibility when developing and assessing these model ensembles.

In addition to evaluating current models, the CNI knowledge base can be a separate launching pad for building models to find CNIs directly. One distinct advantage of a CNI knowledge base is that the database will continuously grow as each new site's audit results are added to the database. A detailed analysis of the CNI knowledge base is beyond the scope of this thesis, but would provide adequate material for additional research as another thesis topic.

One of the main reasons for developing a CNI knowledge base is the paucity of readily available, documented cases of fraud within DFAS. There are likely a sizeable number of cases available at investigative agencies, which for various reasons have not made it into IR Seaside's database. Even though CNIs provide additional knowledge, if the main goal is to find fraud, then the best source of knowledge should still be a comprehensive fraud data warehouse. The current fraud data has essentially remained unchanged for three years. This issue has been touched upon in Section III.C, but is another area for research that could provide an excellent thesis opportunity.

3. Constructing Ensembles from Current Library.

IR Seaside has assembled a large library of models over the last few years. We believe there are enough models to begin combining them without having to repeat the model building process for each audit. Not only can ensembles be built using the current library, but also record selection can be improved by using alternatives to the majority-voting scheme. As shown in Section V.C, the best majority-vote ensemble CNI classification rate for the four sites was 70.7%. The ensemble performance may be improved by treating the model votes as a classification problem. Using the same data and models, a neural net was able to achieve an increase in correct classification rate of 22% over the majority-vote scheme. This increased performance demonstrates that the concept is valid and alternatives to majority voting should be explored and implemented.

4. Model Development Using CNI Knowledge Base.

Differences exist between the fraud and CNI knowledge bases such as population prevalences, the use of site data for training models, and data set sizes. However, the majority of the concepts such as model building/testing, model evaluation and ensemble building will remain the same. The data-mining team should be able to assimilate the CNI knowledge base into the data-mining process rather quickly. If further analysis is conducted on the CNI knowledge base, new insights may come forth that show previously undiscovered relationships within the data. This is an area ripe for further research, whether as a

thesis or as data mining by the IR Seaside team. Whatever the means, the team now has another tool available when searching for fraud within DFAS payments.

C. CLOSING REMARKS.

IR Seaside is using modern technology to improve the audit process. It must continue to explore new technologies and methodologies in its effort to save the taxpayers' money from fraudulent activity and misappropriation. The ideas presented in this thesis developed from daily interactions with modelers, auditors and supervisors. This thesis is hopefully the first step in a lasting relationship between the Naval Postgraduate School and DFAS IR Seaside, because there are many interesting concepts that have been touched upon here. The opportunity for thesis students to bring in fresh ideas and capabilities will only add to the value of the data-mining efforts, while providing exciting research opportunities to the students and the data-mining team alike.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. S-PLUS CODE FOR CHAPTER IV ANALYSIS

This appendix displays the S-Plus code used for the analyses performed in Chapter 4. It is provided for the reader to see the details of each analysis. The S-Plus commands are highlighted in boldface.

1 – Supervised versus random selection (Section IV.B).

This section shows the S-Plus analysis of the Mantel-Haenzel test to test the hypothesis that supervised modeling is more effective at identifying CNIs than random selection. The hypothesis is shown on page 29.

We initially need to build the array of results from the four sites using the data shown in Table 1.

```
> SvR_array(0,c(2,2,4))
> SvR[,,1]_c(169,63,52,40)
> SvR[,,2]_c(69,7,118,23)
> SvR[,,3]_c(16,13,42,37)
> SvR[,,4]_c(53,16,44,34)
> SvR

, , 1
     [,1] [,2]
[1,]  169   52
[2,]   63   40

, , 2
     [,1] [,2]
[1,]   69  118
[2,]    7   23

, , 3
     [,1] [,2]
[1,]   16   42
[2,]   13   37

, , 4
     [,1] [,2]
[1,]   53   44
[2,]   16   34
```

The next step is to build the row (ri), column (ci), site record number (Ni) and positively identified CNI (xi) variables for calculating test statistic as shown in Table 2
.

```
> ri_c(169+52,69+118,16+42,53+44)
```

```
> ri
[1] 221 187  58  97

> ci_c(169+63,69+7,16+13,53+16)
> ci
[1] 232  76  29  69

> xi_SvR[1,1,]
> xi
[1] 169  69  16  53

> Ni_c(sum(SvR[,,1]),sum(SvR[,,2]),sum(SvR[,,3]),sum(SvR[,,4]))
> Ni
[1] 324 217 108 147
```

The next step is to calculate the test statistic Z.

```
> Z.numer_sum(xi)-sum(ri*ci/Ni)
> Z.numer
[1] 22.16

> Z.denom_sqrt(sum(ri*ci*(Ni-ri)*(Ni-ci)/(Ni^3)))
> Z.denom
[1] 5.80

> Z.stat_Z.numer/Z.denom
> Z.stat
[1] 3.82
```

The final step is to determine the p-value by comparing Z.stat to a standard normal distribution.

```
> 1-pnorm(T5)
[1] 0.000067
```

The result is a resounding rejection of the null hypothesis with the p-value = 0.000067.

2 - Model improvement test (Section IV.C).

This section tests the hypothesis that the model scores are not increasing from site to site over time. The hypothesis is shown on page 33.

Start by building the data matrix shown in Table 3.

```
> SD5.scores_c(552.93,552.93,533,529.98,528.85)
> SAI5.scores_c(536.53,535.38,545.05,565.98,534.22)
> SAC5.scores_c(539.84,530.09,550.19,558.47,568.78)
> OA7.scores_c(553.07,530.63,589.67,558.46,584.96,547.19,545.70)
```

```
> Mod.names_c(rep("SD",5),rep("SAI",5),rep("SAC",5), rep("OA",7))
> Mod.scores_c(SD5.scores, SAI5.scores, SAC5.scores, OA7.scores)
> Mod.imp_data.frame(Mod.names, Mod.scores)

> Mod.imp
   Mod.names Mod.scores
 1        SD     552.93
 2        SD     552.93
 3        SD     533.00
 4        SD     529.98
 5        SD     528.85
 6       SAI     536.53
 7       SAI     535.38
 8       SAI     545.05
 9       SAI     565.98
10       SAI     534.22
11       SAC     539.84
12       SAC     530.09
13       SAC     550.19
14       SAC     558.47
15       SAC     568.78
16        OA     553.07
17        OA     530.63
18        OA     589.67
19        OA     558.46
20        OA     584.96
21        OA     547.19
22        OA     545.70

Build an ANOVA and check if it meets assumptions
> Mod.imp.aov_aov(Mod.scores~Mod.names, data=Mod.imp)
> Mod.imp.aov
Call:
   aov(formula = Mod.scores ~ Mod.names, data = Mod.imp)

Terms:
                Mod.names Residuals
 Sum of Squares  1237.267  5007.780
Deg. of Freedom         3        18

Residual standard error: 16.67963
Estimated effects may be unbalanced

> summary(Mod.imp.aov)
          Df Sum of Sq  Mean Sq  F Value     Pr(F)
Mod.names  3  1237.267 412.4223 1.482414 0.2529318
Residuals 18  5007.780 278.2100

> plot(Mod.imp.aov)
```

ANOVA model plots (not shown) show that ANOVA assumptions are not met. The next step is to build the rank matrix to evaluate the data with a Kruskal-Wallis test.

```
> table(rank(Mod.scores), Mod.names)
     OA SAC SAI SD
   1  0   0   0  1
   2  0   0   0  1
   3  0   1   0  0
   4  1   0   0  0
   5  0   0   0  1
   6  0   0   1  0
   7  0   0   1  0
   8  0   0   1  0
   9  0   1   0  0
  10  0   0   1  0
  11  1   0   0  0
  12  1   0   0  0
  13  0   1   0  0
14.5  0   0   0  2
  16  1   0   0  0
  17  1   0   0  0
  18  0   1   0  0
  19  0   0   1  0
  20  0   1   0  0
  21  1   0   0  0
  22  1   0   0  0
```

Perform a Kruskal-Wallis test in S-Plus.

```
> kruskal.test(Mod.scores, Mod.names)

     Kruskal-Wallis rank sum test

data:  Mod.scores and Mod.names
Kruskal-Wallis chi-square = 4.12, df = 3, p-value = 0.25
alternative hypothesis: two.sided
```

To verify the Kruskal-Wallis test, calculate the T statistic shown in Chapter IV.

```
> rank(Mod.scores[Mod.names=="SD"])
[1] 4.5 4.5 3.0 2.0 1.0

> rank(Mod.scores)[Mod.names=="SD"]
[1] 14.5 14.5  5.0  2.0  1.0

> rank(Mod.scores)[Mod.names=="SAI"]
[1]  8  7 10 19  6
```

```
> rank(Mod.scores)[Mod.names=="SAC"]
[1]  9  3 13 18 20

> rank(Mod.scores)[Mod.names=="OA"]
[1] 16  4 22 17 21 12 11

> Ri_c(0,0,0,0)
> Ri[1]_sum(rank(Mod.scores)[Mod.names=="SD"])
> Ri[2]_sum(rank(Mod.scores)[Mod.names=="SAI"])
> Ri[3]_sum(rank(Mod.scores)[Mod.names=="SAC"])
> Ri[4]_sum(rank(Mod.scores)[Mod.names=="OA"])
> ni_c(5,5,5,7)
> N_22
> T.imp_12*sum((Ri^2)/ni)/(N*(N+1))-3*(N+1)
> T.imp
[1] 4.12
```

The final step is to check the test statistic against a chi-squared distribution with three degrees of freedom.

```
> 1-pchisq(T.imp, 3)
[1] 0.25
```

The result is a p-value of 0.25, which shows that the data does not provide sufficient evidence to reject the null hypothesis.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. CLEMENTINE IMPLEMENTATION OF MODEL DEVELOPMENT SCORE FUNCTION

This appendix shows the implementation of the scoring function within Clementine. Screenshots that show nodes within the scoring function will be presented and explained. This provides DFAS IR with the information necessary to make informed changes to the scoring stream if they deem it necessary in the future.

Figure 7 is the entire scoring function stream. The stream starts with a filter node that removes all fields except for "FRAUD_TYPE", "CASE_NAME" and "$N-FRAUD_TYPE". Each node will be described and screenshots of specific nodes will help explain the internal code. The data used for this screenshot is for illustrative purposes and comes from the actual voting data from a recent training dataset.

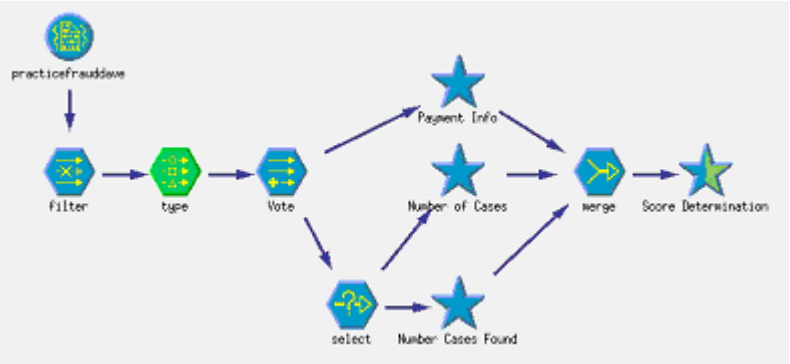


Figure 7. Clementine Score Development Screenshot

The first type node ensures the three fields are all typed as sets for proper calculations. The derive node, "Vote" changes the "$N-FRAUD_TYPE" to a 0/1 flag. Depending on the model type used, this node will rename the predicted fraud type into the "Vote" flag to enable the downstream nodes to properly work. For DFAS purposes, separate scoring streams will be needed for use with each type of model that will be used (i.e. C5.0 or NN). After the "Vote" node, the data is split into two sub streams for determination of the payment and case variables.

The "Payment Info" Supernode is shown in Figure 8. This node contains the stream that breaks down the payment information into the contingency table data of Table 5. The "ActFraud" node creates a 0/1 flag that identifies a record as actual fraud or nonfraud. The next four derive nodes tally the number of records that would fall within

each cell in Table 5. The sort node then sorts the records from highest “a” value to lowest, which means the first record has the sum for each of the four cell values “a”, “b”, “c”, and “d”. The filter node removes all unnecessary fields, and the “Extract Data” node keeps only the first record, which contains the Table 5 cell values. The last type node changes all the fields to integers for calculation purposes.

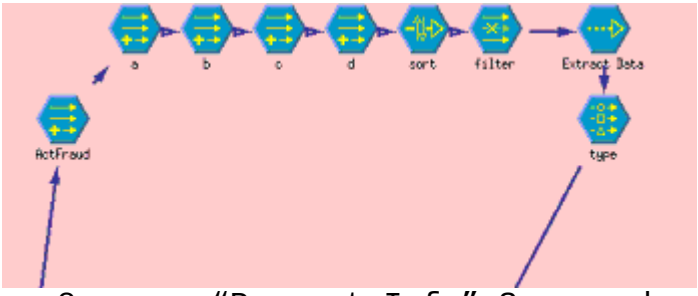


Figure 8. “Payment Info” Supernode

The other branch following the “Vote” node in Figure 7 goes to the case sub stream. The first node is a select node that keeps only the fraud data and strips off all non-fraud data. From this node, two sub streams are formed, one to determine the total number of cases and the other to determine the number of cases identified as fraud.

Figure 9 shows the “Number of Cases” Supernode. The first node in this branch is an aggregate node that counts each “CASE_NAME” in the dataset. This is followed by a derive node, “NumCases,” which tallies the number of cases. The sort node then sorts by number of cases from highest number to lowest. This sort places the case total in the first record location, to subsequently be stripped off by the “1” sample node. The final filter node removes all fields except for “NumCases.”

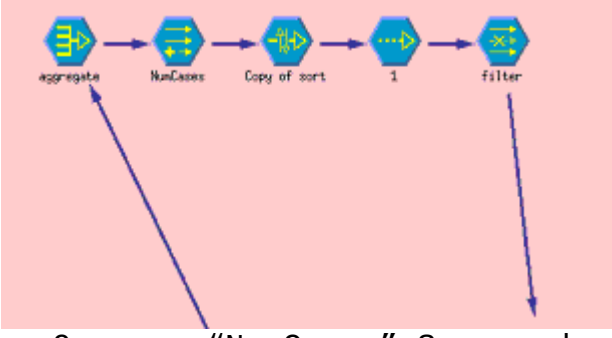


Figure 9. “NumCases” Supernode

The other Supernode in the case sub stream is shown in Figure 10. This Supernode is similar to “NumCases” except that a select node at the beginning of the stream keeps only the records that received a “Vote” of 1. The remainder of the stream works the same way as “NumCases.” The result is that if any payments within a case were classified as potential fraud, then that case is identified. The output from this Supernode is the field “NumCasesFound.”

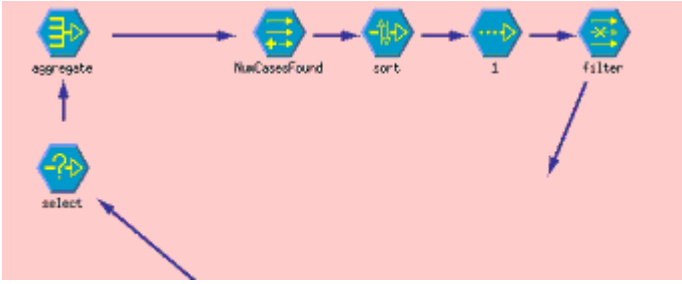


Figure 10. “NumCasesFound” Supernode

Following the three Supernodes, all the fields are merged together for the score determination. The fields that now remain are “a”, “b”, “c”, “d”, “NumCases” and “NumCasesFound.” This data is fed into the “Score Determination” Supernode shown in Figure 11.

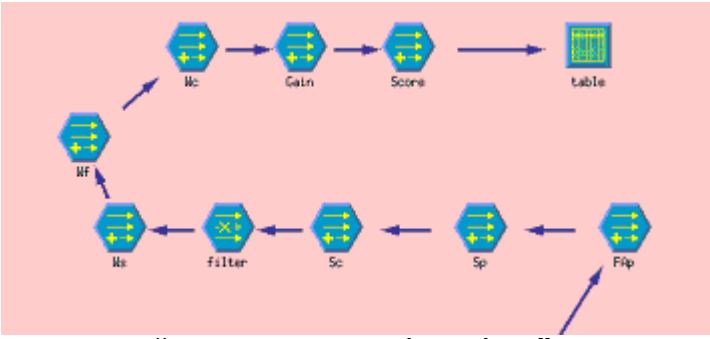


Figure 11. “Score Determination” Supernode

The first three nodes in Figure 11 calculate the payment False Alarm (FAp), payment Sensitivity (Sp), and the case Sensitivity (Sc). The first two calculations come straight from Equations 6 and 7 on page 38. The case sensitivity is found by dividing the “NumCasesFound” by “NumCases.” The filter node then removes unneeded fields. The next four nodes are used to assign the weights used in the score function. If DFAS desires to change weights in the future then it will be a simple matter to just reassign the weights within the derive nodes. The final derive node

is the actual score calculation for the data run as shown in Equation 18 on page 47. Figure 12 shows the implementation of the equation in the derive node.

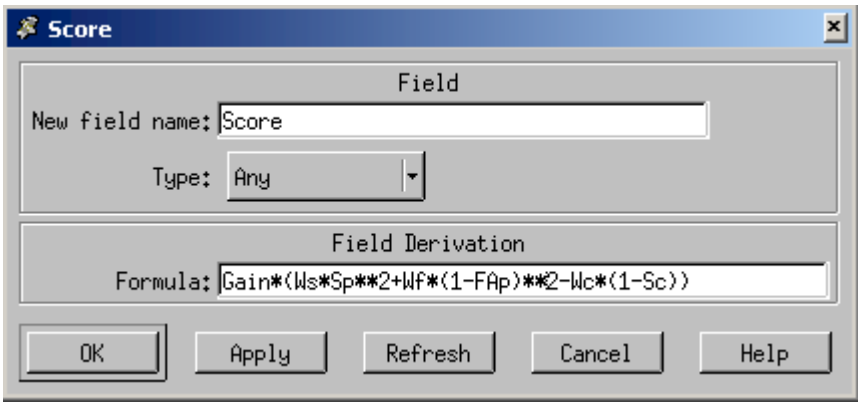
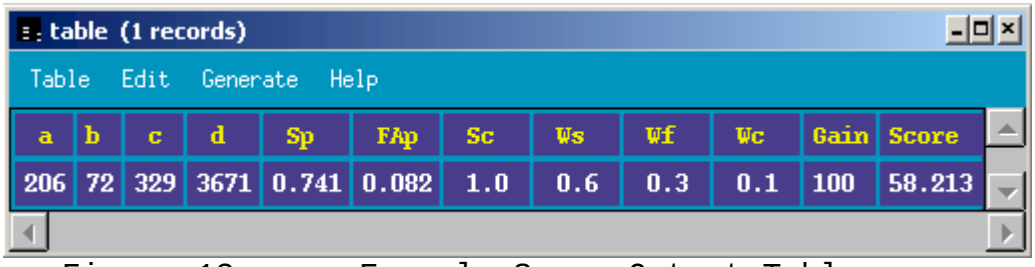


Figure 12. "Score" Node Screenshot

The final node in the "Score Determination" Supernode is a table to display the results of the score determination. The four cell values of Table 5 are displayed, as are the pertinent sensitivity rates and false alarm rates. Additionally the weights and score are displayed. The resulting output is shown in Figure 13. It is then a simple matter to highlight the information in this table and copy it to an Excel spreadsheet for model comparisons.



a	b	c	d	Sp	FAp	Sc	Ws	Wf	Wc	Gain	Score
206	72	329	3671	0.741	0.082	1.0	0.6	0.3	0.1	100	58.213

Figure 13. Example Score Output Table

APPENDIX C. CNI KNOWLEDGE BASE FIELDS

This appendix consists of a table of variable fields available across the four sites of OA, SD, SAC and SAI. The locations of each field are shown next to the name with a (1) meaning that field was represented at the site. Additionally, whether or not that field was retained in the CNI knowledge base is also represented by a (0) or (1) in the status column. Finally, any comments regarding each field are recorded to show the pertinent information regarding that field.

Field Name	OA	SAC	SAI	SD	Status	Comments
ADDRESS13	1	1	1	1	1	
AGGREG_ADR	1	1	1	1	1	
AGGREG_PAYEE	1	1	1	1	1	
ALL_OTHER	1	1	1	1	1	
ALLX	1	1	1	1	1	
AVG_5K	1	1	1	1	1	
AUDIT_SITE	0	0	0	0	1	Generated to track which site a record came from.
AWARD_DT	1	1	1	1	1	
BNR_AMT	1	0	0	0	0	Populated with N/As and only at OA.
BCO_ID	0	1	1	1	1	Only populated at SAI. Updated OA.
BRAC	1	1	1	1	1	
C_INV_NUM	1	1	1	1	1	
CAGE_CD	0	1	1	1	1	Update OA, but OA all blank.
CASE	1	0	0	1	0	Changed to CASE_NAME starting at SA, changed SD & OA field name to match.
CASE_NAME	0	1	1	0	1	Updated OA and SD to match.
CDF_REMIT_TO	1	0	0	0	0	Renamed to CDF_RMT_NAME to match later sites.
CDF_RMT_CITY	1	1	1	1	1	
CDF_RMT_L1	1	1	1	1	1	
CDF_RMT_L2	1	1	1	1	1	
CDF_RMT_L3	1	1	1	1	1	
CDF_RMT_L4	1	1	1	1	1	
CDF_RMT_NAME	0	1	1	1	1	OA field CDF_REMIT_TO changed to match.
CDF_RMT_ST	1	1	1	1	1	
CDF_RMT_ZIP	1	1	1	1	1	
CHK_AMT	1	1	1	1	1	

Field Name	OA	SAC	SAI	SD	Status	Comments
CHK_AWARD_DT	1	1	1	1	1	
CHK_CAN_DT	0	1	1	1	1	OA field DTCKCAN changed to match
CHK_DT	1	1	1	1	1	
CHK_INV_DT	1	1	1	1	1	
CHK_INV_RECV_DT	1	1	1	1	1	
CHK_NUM	1	1	1	1	1	
CHK_STAT	1	1	1	1	1	
CHK_XREF	1	1	1	1	1	
CNI_Code	1	1	1	1	1	
CNT_CDF	1	1	1	1	1	
CON_AMT	0	1	1	1	1	Updated OA.
CON_STAT	0	1	1	1	1	Updated OA.
CON_TYP	0	1	1	1	1	Updated OA.
Control_Number	1	1	1	1	1	
DBOF	1	1	1	1	1	
DEL_ORD	1	1	1	1	1	
DFAR	1	0	0	0	0	Changed 1 to 0 and 0 to 1, then renamed NOT_DFAR.
DISC_AMT	1	1	1	1	1	
DISCOUNT	1	1	1	1	1	
DOV_AMT	1	1	1	1	1	
DOVAMT_1K	1	1	1	1	1	
DOVAMT_2K	1	1	1	1	1	
DSSN	1	1	1	1	1	
DTCKCAN	1	0	0	0	0	Changed name to CHK_CAN_DT to match other sites.
DUNS	1	0	0	0	0	Changed name to DUNS_NUM to match other sites.
DUNS_NUM	0	1	1	1	1	OA field DUNS changed to match.
DUPAY109	0	1	1	0	1	Generated new fields in OA and SD and populated with zeros.
DUPAY111	0	1	1	0	1	
DUPPAY102	1	1	1	1	1	
DUPPAY110	1	1	1	1	1	
EFT_ACCT	1	1	1	1	1	
EFT_ADR	1	1	1	1	1	
EFT_PAYEE	1	1	1	1	1	
EFT_RTN	1	1	1	1	1	
EFT_TRAN	1	0	0	0	0	Populated with N/As and only at OA, deleted.
EFT_TRANS	1	0	0	0	0	Populated with all zeros and only at OA, deleted.
ENHANCE_PAYEE	1	1	1	1	1	
FCUR_TYP	1	0	0	0	0	Populated with all zeros and only at OA, deleted.
FEW_PYMT	1	1	1	1	1	
FILE_SEQ	1	1	1	1	1	

Field Name	OA	SAC	SAI	SD	Status	Comments
FOB	0	1	1	1	1	Updated OA.
FRAUD_TYP	1	0	0	0	0	All these records are Nonfraud records, fields of no value, deleted.
FRAUD_TYPE	1	1	1	1	0	
FRT_AMT	1	1	1	1	1	
FRT_STAT	1	1	1	1	1	
INDEBT	1	0	0	0	0	Populated with all N/As and only at OA, deleted.
GS_IND	0	1	1	1	1	Updated OA.
INT_PD_AMT	0	1	1	1	1	OA field IP_AMT changed to match.
INTEREST	1	1	1	1	1	
INV_AMT	1	1	1	1	1	
INV_AWARD_DT	1	1	1	1	1	
INV_CNT	1	1	1	1	1	
INV_DT	1	1	1	1	1	
INV_ENTR_DT	0	1	1	1	1	Updated OA.
INV_NUM	1	1	1	1	1	
INV_PAYEE	1	1	1	1	1	
INV_RCVD	1	1	1	1	1	
INV_RECV_AWARD_DT	1	1	1	1	1	
INV_RECV_INV_DT	1	1	1	1	1	
INV_SEQ	1	1	1	1	1	
INV_SUPP	1	0	0	0	0	Populated with blanks and only at OA, deleted.
IP_AMT	1	0	0	0	0	Changed name to INT_PD_AMT to match other sites.
JON	1	0	0	0	0	Populated with blanks and only at OA, deleted.
LINEITEM	1	1	1	1	1	
LOCKBOX	1	1	1	1	1	
LOST_AMT	1	1	1	1	1	
LOST_CD	1	1	1	1	1	
M_PYMT	1	1	1	1	1	
MAN_IND	1	1	1	1	1	
MDSE_ACC_DT	0	1	1	1	1	Updated OA.
MDSE_DEL_DT	0	1	1	1	1	Updated OA.
MILPAY	1	1	1	1	1	
MISC_CHG	1	0	0	0	0	Populated with all N/As and only at OA, deleted.
MISC_OBLIG	1	1	1	1	1	
MULTI_ADDR_K	1	1	1	1	1	
MULTI_ADR	1	1	1	1	1	
MULTI_EFT_K	1	1	1	1	1	
MULTI_PAYEE	1	1	1	1	1	
MULTI_PAYEE_K	1	1	1	1	1	
MULTI_PAYTIN	1	1	1	1	1	
MULTI_TINS	1	1	1	1	1	
MULTI_TINS_K	1	1	1	1	1	
NET_VND	0	1	1	1	1	Updated OA.

Field Name	OA	SAC	SAI	SD	Status	Comments
NEW_ID	1	0	0	0	0	ID numbers only for OA, no value
NOT_DFAR	0	1	1	1	1	OA Updated from DFAR
NUM_EE_K	0	1	1	0	1	Transforms only available starting with SA
NUMADR_K	0	1	1	0	1	
NUMADREE	0	1	1	0	1	
NUMEFT_K	0	1	1	0	1	
NUMEFTEE	0	1	1	0	1	
DOV_NUM	1	0	0	0	1	Removed each site prefix.
seq_id	1	0	0	0	1	
OA_IND	1	0	0	0	0	Populated with O, field of no value, deleted.
ORDER_CDF	1	1	1	1	1	
ORDER_TO_PAY	1	0	0	0	0	Only available at OA, no longer used, deleted.
OTHERX	1	1	1	1	1	
OVER_BLD	1	0	0	0	0	Populated with all N/As and only at OA, deleted.
PAY_CMRA_ADR	1	0	0	0	0	Populated with all zeros and only at OA, deleted.
PAY_ORDER	1	1	1	1	1	
PAYEE	1	1	1	1	1	
PAYEE_4_PYMT	1	1	1	1	1	
PAYEE13	1	1	1	1	1	
PAYMENT	1	1	1	1	1	
PIIN	1	1	1	1	1	
PMT_CAT	1	1	1	1	1	
PMT_FREQ_HI	1	1	1	1	1	
PMT_FREQ_LO	1	1	1	1	1	
PMT_METH	1	1	1	1	1	
PMT_METH_C	1	0	0	0	0	Info contained within PMT_METH, extra flags not required, deleted.
PMT_METH_D	1	0	0	0	0	
PMT_METH_E	1	0	0	0	0	
PMT_METH_I	1	0	0	0	0	
PMT_METH_NULL	1	0	0	0	0	
PMT_METH_P	1	0	0	0	0	
PMT_METH_R	1	0	0	0	0	
PMT_METH_X	1	0	0	0	0	
PMT_NUM	1	1	1	1	1	
PMT_PROV	1	1	1	1	1	
PMT_PROV_A	1	0	0	0	0	Info contained within PMT_PROV, extra flags not required, deleted.
PMT_PROV_F	1	0	0	0	0	
PMT_PROV_H	1	0	0	0	0	
PMT_PROV_NULL	1	0	0	0	0	

Field Name	OA	SAC	SAI	SD	Status	Comments
PMT_PROV_P	1	0	0	0	0	Info contained within PMT_PROV, extra flags not required, deleted.
PMT_PROV_R	1	0	0	0	0	
PMT_TYPE	1	1	1	1	1	
PMT_TYPE_C	1	0	0	0	0	Info contained within PMT_TYPE, extra flags not required, deleted.
PMT_TYPE_F	1	0	0	0	0	
PMT_TYPE_NULL	1	0	0	0	0	
PMT_TYPE_P	1	0	0	0	0	
POBOX	1	1	1	1	1	
PPA_XMPT	0	1	1	1	1	Updated OA, but it is all blanks.
Pull_Voucher	1	1	1	1	0	All records are Pull_Voucher, deleted.
REMIT_S	1	0	0	0	1	Populated with all blanks and only at OA, deleted.
REMIT_TO	1	0	0	0	0	Changed name to RMT_NAME to match other sites.
RMT_CD	0	1	1	1	1	Updated OA, but it is all blanks.
RMT_CITY	1	1	1	1	1	
RMT_L1	1	1	1	1	1	
RMT_L2	1	1	1	1	1	
RMT_L3	1	1	1	1	1	
RMT_L4	1	1	1	1	1	
RMT_NAME	0	1	1	1	1	OA Updated from REMIT_TO.
RMT_ST	1	1	1	1	1	
RMT_ZIP	1	1	1	1	1	
RNDM_NUM	1	1	1	1	0	Random number for record selection, deleted.
SITE_ID	1	1	1	1	1	
STE	1	1	1	1	1	
SUB_DT	1	1	1	1	1	
SYS_DCN	1	1	1	1	1	
SYS_ID	1	1	1	1	1	
SYS_UNQ	1	1	1	0	1	Missing at SD, appears to only be used at SAI, SD all blanks.
TAX_AMT	1	1	1	1	1	
TEST1	0	1	1	0	0	Only used at SA, populated with all zeros, deleted.
TEST2	0	1	1	0	0	
TEST3	0	1	1	0	0	
TEST4	0	1	1	0	0	
TEST5	0	1	1	0	0	
TEST6	0	1	1	0	0	
TIN	1	1	1	1	1	
TINS	1	1	1	1	1	
TRANS_NUM	1	1	1	1	1	
UNUSUAL	1	1	1	1	1	
VE1_CD	0	1	1	1	1	Updated OA.
VE2_CD	0	1	1	1	1	Updated OA.
VE3_CD	0	1	1	1	1	Updated OA.
VE4_CD	0	1	1	1	1	Updated OA.
VE5_CD	0	1	1	1	1	Updated OA.

Field Name	OA	SAC	SAI	SD	Status	Comments
VE_PMT	1	0	0	0	0	Same as VE4_CD, deleted.
VND_ADR1	0	1	1	1	1	Updated OA.
VND_ADR2	0	1	1	1	1	Updated OA.
VND_ADR3	0	1	1	1	1	Updated OA.
VND_CITY	0	1	1	1	1	Updated OA.
VND_CRED	1	0	0	0	0	Populated with all N/As and only at OA, deleted.
VND_ID	0	1	1	1	1	Updated OA.
VND_NAME	0	1	1	1	1	Updated OA.
VND_ST	0	1	1	1	1	Updated OA.
VND_TYP	0	1	1	1	1	Updated OA.
VND_ZIP	0	1	1	1	1	Updated OA.
VOU_ST_B	1	0	0	0	0	Info contained within VOU_STAT, extra flags not required, deleted.
VOU_ST_D	1	0	0	0	0	
VOU_ST_Q	1	0	0	0	0	
VOU_ST_S	1	0	0	0	0	
VOU_ST_V	1	0	0	0	0	
VOU_STAT	1	1	1	1	1	
VPR_AMT	1	0	0	0	0	Populated with all N/As and only at OA, deleted.
Y1_CUR	1	1	1	1	1	
Y1_PRIOR	1	1	1	1	1	
Y2_CUR_1ST	1	1	1	1	1	
Y2_CUR_2ND	1	1	1	1	1	
Y2_PRIOR	1	1	1	1	1	
Y3_PLUS	1	1	1	1	1	
d	1	1	1	1	1	Duplicate flag.
r	1	1	1	1	1	Random flag.
s	1	1	1	1	1	Supervised flag.
u	1	1	1	1	1	Unsupervised flag.

APPENDIX D. FOUR SITES' MODEL NAMES AND CNI CLASSIFICATION RATES

GAMS#	Model	Model Name	Model Classification Rate
1	SD9	kh_sup_9_nn_1_e10	50.8%
2	SD17	bullet_sup_8_c5_2_pe0	49.9%
3	SD19	kh_sup_3_nn_1_sd1	58.4%
4	SD20	cd_sup_1_c5_2_sd1	46.2%
5	SD21	jg_sup_4_nn_2_sd1	59.7%
6	SD22	tc_sup_2_nn_2_sd1	51.2%
7	SD23	tc_sup_8_c5_2_sd1	51.5%
8	SAI1	tc_sup_9_nn_3_oa1_sa1iaps	56.0%
9	SAI2	kh_sup_5_c5_5_oa1_sa1iaps	48.3%
10	SAI3	kh_sup_2_c5_oa1_sa1iaps	50.6%
11	SAI4	kh_sup_11_nn_1_e10_sa1iaps	52.9%
12	SAI5	cd_sup_11_c5_2_oa1_sa1iaps	52.4%
13	SAI6	kh_sup_10_c5_2_oa1_sa1iaps	54.5%
14	SAI7	cd_4_nn_1_sa1iaps	57.7%
15	SAI8	rtf_1_c5_5_sa1iaps	54.7%
16	SAI9	rtf_3_xx_3_sa1iaps	56.3%
17	SAI10	rtf_6_xx_1_sa1iaps	45.1%
18	SAI11	dr_4_dyna_1_sa2iaps	52.6%
19	SAC1	tc_sup_9_nn_3_oa1_sa1caps	55.1%
20	SAC2	kh_sup_5_c5_5_oa1_sa1caps	51.9%
21	SAC3	kh_sup_2_c5_oa1_sa1caps	55.2%
22	SAC4	kh_sup_11_nn_1_e10_sa1caps	52.9%
23	SAC5	cd_sup_11_c5_2_oa1_sa1caps	54.2%
24	SAC6	kh_sup_10_c5_2_oa1_sa1caps	52.8%
25	SAC7	rtf_3_xx_2_sa1caps	52.2%
26	SAC8	rtf_1_c5_5_sa1caps	44.9%
27	SAC9	cd_4_nn_1_sa1caps	51.7%
28	SAC10	cd_7_cd_2_sa1caps	52.9%
29	SAC11	cd_sup_1_nn_4_sa1caps	50.4%
30	OA1	kh_sup_6_nn_5_oa1	48.8%
31	OA2	kh_sup_7_br_5_oa1	63.1%
32	OA3	kh_sup_10_c5_2_oa1	64.5%
33	OA4	kh_sup_5_c5_5_oa1	60.0%
34	OA5	kh_sup_2_c5_5_oa1	52.8%
35	OA6	tc_sup_9_nn_3_oa1	57.5%
36	OA7	cd_sup_11_c5_2_oa1	49.0%
37	OA8	kh_sup_11_nn_1_e10	54.4%
38	OA9	kh_sup_9_nn_1_e10	51.5%
39	OA10	cd_sup_6_nn_2_e10	49.9%
40	OA11	cd_sup_7_c5_2_e10	56.8%
41	OA12	cd_sup_4_nn_4_e10	56.0%
42	OA13	cd_sup_1_c5_1_pe0	49.6%
43	OA14	rtf_sup_2_nn_6_pe0	55.1%
44	OA15	kirby_sup_3_c5_1_pe0	58.4%
45	OA16	yoshi_sup_4_nn_1_pe0	56.1%
46	OA17	peach_sup_5_br_2_pe0	54.7%
47	OA18	kf_sup_6_nn_1_pe0	49.6%
48	OA19	bullet_sup_6_c5_2_pe0	44.2%
49	OA20	bullet_sup_8_c5_2_pe0	50.8%
50	OA21	kf_sup_9_c5_4_pe0	51.7%
51	OA22	kirby_sup_10_nn_6_pe0	50.8%
52	OA23	cd_sup_11_c5_1_pe0	53.3%

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX E. GAMS CODE FOR OPTIMIZED ENSEMBLE

The code presented here is the GAMS formulation and solution for determining the optimized model stream using a majority-vote. There are solutions for model sizes ranging from 3 to 21 models in an ensemble. GAMS model numbers and their respective names and classification rates are shown in Appendix D.

```
-----------------------------------------------------------------------
$TITLE Fraud 02.01.30

$INLINECOM { }
OPTIONS
         SOLPRINT =   OFF,
         DECIMALS =     1,
         LIMCOL   =     0,
         LIMROW   =     0,
         RESLIM   =  750, {max seconds}
         ITERLIM  = 500000, {max pivots}
         OPTCR    =  0.10, {relative integrality tolerance}
         RMIP     =  cplex,
         MIP      =  osl ; {OSL, CPLEX, XA, ... }

 SETS  r           "records"
     / r001*r563 /

       m           "models"
     / m001*m052 /

       cni(r)      "audited records found fraudulent"
     /
$INCLUDE cni.dat
     /
       vote(r,m)  "fraud indications"
     /
$INCLUDE vote.dat
     /;

 PARAMETER audit(r), agree(r,m) ;

 LOOP(r,
   audit(r) = 0 ;
 );
 LOOP(cni(r),
   audit(r) = 1 ;
 );
 LOOP((r,m),
   agree(r,m) = 0 ;
 );
 LOOP(vote(r,m),
   agree(r,m) = 1 ;
 );
 LOOP((r,m),
```

```
  IF( agree(r,m)=audit(r),
     agree(r,m) = 1 ;
  ELSE
     agree(r,m) = 0 ;
  );
);

SCALAR enssize , majvotes ;

VARIABLES
  Z                  fraudulent records found by selected ensemble ;

BINARY VARIABLES
  CLASSIFY(r)
  SELECT(m)
;

EQUATIONS
  OBJECT             objective function
  MAJORITY(r)        majority vote required to classify any record
  ENSEMBLESZ         restrict maximum ensemble size
  ;

OBJECT..
  Z =e= SUM(r,CLASSIFY(r))
;

MAJORITY(r)..
  majvotes*CLASSIFY(r) - SUM(m$agree(r,m),SELECT(m)) =l= 0
;

ENSEMBLESZ..
  SUM(m,SELECT(m)) =l= enssize
;
MODEL FRAUD / ALL /;

For (enssize= 3 to 21 by 2,
  majvotes = FLOOR(enssize/2) + 1 ;
  SOLVE FRAUD USING MIP MAXIMIZING  Z ;
  DISPLAY enssize
  DISPLAY SELECT.l   ;
  );
----------------------------------------------------------------------
```

```
LOOPS                            FOR/WHILE   1

MODEL STATISTICS
BLOCKS OF EQUATIONS       3     SINGLE EQUATIONS      565
BLOCKS OF VARIABLES       3     SINGLE VARIABLES      616
NON ZERO ELEMENTS     16727     DISCRETE VARIABLES    615

GENERATION TIME      =        0.063 SECONDS    2.4 Mb      WIN200-121
EXECUTION TIME       =        0.297 SECONDS    2.4 Mb      WIN200-121

               S O L V E      S U M M A R Y
     MODEL   FRAUD               OBJECTIVE  Z
     TYPE    MIP                 DIRECTION  MAXIMIZE
     SOLVER  OSL                 FROM LINE  18470

**** SOLVER STATUS     2 ITERATION INTERRUPT
**** MODEL STATUS      8 INTEGER SOLUTION
**** OBJECTIVE VALUE              391.0000

 RESOURCE USAGE, LIMIT        386.391       750.000
 ITERATION COUNT, LIMIT    500012        500000

 OSL Version 1 Mar 21, 2001 WIN.OS.O2 20.0 058.043.039.WAT
 Work space allocated           --    3.65 Mb

 Relaxed optimum objective value:     460.15538
 Bound on best integer solution:      441.25000
 Objective value of this solution:    391.00000

 Relative gap: .11388 Absolute gap:   50.250000
 Optcr       : .10000 Optca:          0.0

 The solution does not satisfy the termination tolerances

**** REPORT SUMMARY :        0     NONOPT
                             0 INFEASIBLE
                             0  UNBOUNDED

E x e c u t i o n
----  18471 PARAMETER enssize              =          3.0

----  18472 VARIABLE  SELECT.L

m021 1.0,    m031 1.0,    m032 1.0
```

```
LOOPS                            FOR/WHILE   2

MODEL STATISTICS
BLOCKS OF EQUATIONS       3     SINGLE EQUATIONS      565
BLOCKS OF VARIABLES       3     SINGLE VARIABLES      616
NON ZERO ELEMENTS     16727     DISCRETE VARIABLES    615

GENERATION TIME      =        0.079 SECONDS    2.4 Mb      WIN200-121
EXECUTION TIME       =        0.235 SECONDS    2.4 Mb      WIN200-121

               S O L V E      S U M M A R Y
     MODEL   FRAUD               OBJECTIVE  Z
     TYPE    MIP                 DIRECTION  MAXIMIZE
     SOLVER  OSL                 FROM LINE  18470

**** SOLVER STATUS     2 ITERATION INTERRUPT
**** MODEL STATUS      8 INTEGER SOLUTION
**** OBJECTIVE VALUE              391.0000

 RESOURCE USAGE, LIMIT        388.512       750.000
 ITERATION COUNT, LIMIT    500033        500000

 OSL Version 1 Mar 21, 2001 WIN.OS.O2 20.0 058.043.039.WAT
 Work space allocated           --    3.65 Mb

 Relaxed optimum objective value:     486.35442
 Bound on best integer solution:      473.76190
 Objective value of this solution:    391.00000

 Relative gap: .17469 Absolute gap:   82.761905
 Optcr       : .10000 Optca:          0.0

 The solution does not satisfy the termination tolerances

**** REPORT SUMMARY :        0     NONOPT
                             0 INFEASIBLE
                             0  UNBOUNDED

E x e c u t i o n
----  18471 PARAMETER enssize              =          5.0

----  18472 VARIABLE  SELECT.L

m021 1.0,    m031 1.0,    m032 1.0,    m034 1.0,    m043 1.0
```

```
LOOPS                              FOR/WHILE   3

MODEL STATISTICS
BLOCKS OF EQUATIONS       3     SINGLE EQUATIONS      565
BLOCKS OF VARIABLES       3     SINGLE VARIABLES      616
NON ZERO ELEMENTS     16727     DISCRETE VARIABLES    615

GENERATION TIME      =        0.063 SECONDS    2.4 Mb      WIN200-121
EXECUTION TIME       =        0.266 SECONDS    2.4 Mb      WIN200-121

               S O L V E      S U M M A R Y
     MODEL   FRAUD               OBJECTIVE  Z
     TYPE    MIP                 DIRECTION  MAXIMIZE
     SOLVER  OSL                 FROM LINE  18470

**** SOLVER STATUS     2 ITERATION INTERRUPT
**** MODEL STATUS      8 INTEGER SOLUTION
**** OBJECTIVE VALUE              398.0000

 RESOURCE USAGE, LIMIT        368.480       750.000
 ITERATION COUNT, LIMIT    500054        500000

 OSL Version 1 Mar 21, 2001 WIN.OS.O2 20.0 058.043.039.WAT
 Work space allocated           --    3.65 Mb

 Relaxed optimum objective value:     497.38410
 Bound on best integer solution:      491.60000
 Objective value of this solution:    398.00000

 Relative gap: .19040 Absolute gap:   93.600000
 Optcr       : .10000 Optca:          0.0

 The solution does not satisfy the termination tolerances

**** REPORT SUMMARY :        0     NONOPT
                             0 INFEASIBLE
                             0  UNBOUNDED

E x e c u t i o n
----  18471 PARAMETER enssize              =          7.0

----  18472 VARIABLE  SELECT.L

m021 1.0,   m030 1.0,   m031 1.0,   m032 1.0,   m033 1.0,   m043 1.0
m049 1.0
```

```
LOOPS                              FOR/WHILE   4

MODEL STATISTICS
BLOCKS OF EQUATIONS       3     SINGLE EQUATIONS      565
BLOCKS OF VARIABLES       3     SINGLE VARIABLES      616
NON ZERO ELEMENTS     16727     DISCRETE VARIABLES    615

GENERATION TIME      =        0.078 SECONDS    2.4 Mb      WIN200-121
EXECUTION TIME       =        0.235 SECONDS    2.4 Mb      WIN200-121

               S O L V E      S U M M A R Y
     MODEL   FRAUD               OBJECTIVE  Z
     TYPE    MIP                 DIRECTION  MAXIMIZE
     SOLVER  OSL                 FROM LINE  18470

**** SOLVER STATUS     2 ITERATION INTERRUPT
**** MODEL STATUS      8 INTEGER SOLUTION
**** OBJECTIVE VALUE              388.0000

 RESOURCE USAGE, LIMIT        338.451       750.000
 ITERATION COUNT, LIMIT    500008        500000

 OSL Version 1 Mar 21, 2001 WIN.OS.O2 20.0 058.043.039.WAT
 Work space allocated           --    3.65 Mb

 Relaxed optimum objective value:     503.25581
 Bound on best integer solution:      499.41429
 Objective value of this solution:    388.00000

 Relative gap: .22309 Absolute gap:   111.41429
 Optcr       : .10000 Optca:          0.0

 The solution does not satisfy the termination tolerances

**** REPORT SUMMARY :        0     NONOPT
                             0 INFEASIBLE
                             0  UNBOUNDED

E x e c u t i o n
----  18471 PARAMETER enssize              =          9.0

----  18472 VARIABLE  SELECT.L

m002 1.0,   m016 1.0,   m029 1.0,   m030 1.0,   m031 1.0,   m032 1.0
m034 1.0,   m037 1.0,   m043 1.0
```

```
 LOOPS                            FOR/WHILE   5

MODEL STATISTICS
BLOCKS OF EQUATIONS       3     SINGLE EQUATIONS      565
BLOCKS OF VARIABLES       3     SINGLE VARIABLES      616
NON ZERO ELEMENTS     16727     DISCRETE VARIABLES    615

GENERATION TIME      =        0.078 SECONDS    2.4 Mb      WIN200-121
EXECUTION TIME       =        0.250 SECONDS    2.4 Mb      WIN200-121

               S O L V E      S U M M A R Y
     MODEL   FRAUD               OBJECTIVE  Z
     TYPE    MIP                 DIRECTION  MAXIMIZE
     SOLVER  OSL                 FROM LINE  18470

**** SOLVER STATUS     2 ITERATION INTERRUPT
**** MODEL STATUS      8 INTEGER SOLUTION
**** OBJECTIVE VALUE              398.0000

 RESOURCE USAGE, LIMIT        351.721       750.000
 ITERATION COUNT, LIMIT    500052        500000

 OSL Version 1 Mar 21, 2001 WIN.OS.O2 20.0 058.043.039.WAT
 Work space allocated           --    3.65 Mb

 Relaxed optimum objective value:     505.85574
 Bound on best integer solution:      502.83333
 Objective value of this solution:    398.00000

 Relative gap: .20849 Absolute gap:   104.83333
 Optcr       : .10000 Optca:          0.0

 The solution does not satisfy the termination tolerances

**** REPORT SUMMARY :        0     NONOPT
                             0 INFEASIBLE
                             0  UNBOUNDED

E x e c u t i o n
----  18471 PARAMETER enssize              =         11.0

----  18472 VARIABLE  SELECT.L

m002 1.0,   m003 1.0,   m029 1.0,   m030 1.0,   m031 1.0,   m032 1.0
m033 1.0,   m034 1.0,   m037 1.0,   m043 1.0,   m047 1.0
```

```
LOOPS                              FOR/WHILE   6

MODEL STATISTICS
BLOCKS OF EQUATIONS       3     SINGLE EQUATIONS      565
BLOCKS OF VARIABLES       3     SINGLE VARIABLES      616
NON ZERO ELEMENTS     16727     DISCRETE VARIABLES    615

GENERATION TIME      =        0.079 SECONDS    2.4 Mb      WIN200-121
EXECUTION TIME       =        0.219 SECONDS    2.4 Mb      WIN200-121

               S O L V E      S U M M A R Y
     MODEL   FRAUD               OBJECTIVE  Z
     TYPE    MIP                 DIRECTION  MAXIMIZE
     SOLVER  OSL                 FROM LINE  18470

**** SOLVER STATUS     2 ITERATION INTERRUPT
**** MODEL STATUS      8 INTEGER SOLUTION
**** OBJECTIVE VALUE              389.0000

 RESOURCE USAGE, LIMIT        346.521       750.000
 ITERATION COUNT, LIMIT    500052        500000

 OSL Version 1 Mar 21, 2001 WIN.OS.O2 20.0 058.043.039.WAT
 Work space allocated           --    3.65 Mb

 Relaxed optimum objective value:     507.02198
 Bound on best integer solution:      505.09524
 Objective value of this solution:    389.00000

 Relative gap: .22985 Absolute gap:   116.09524
 Optcr       : .10000 Optca:          0.0

 The solution does not satisfy the termination tolerances

**** REPORT SUMMARY :        0     NONOPT
                             0 INFEASIBLE
                             0  UNBOUNDED

E x e c u t i o n
----  18471 PARAMETER enssize              =         13.0

----  18472 VARIABLE  SELECT.L

m003 1.0,   m029 1.0,   m030 1.0,   m031 1.0,   m032 1.0,   m033 1.0
m034 1.0,   m036 1.0,   m037 1.0,   m039 1.0,   m043 1.0,   m049 1.0
m051 1.0
```

```
 LOOPS                            FOR/WHILE   7

MODEL STATISTICS
BLOCKS OF EQUATIONS       3     SINGLE EQUATIONS      565
BLOCKS OF VARIABLES       3     SINGLE VARIABLES      616
NON ZERO ELEMENTS     16727     DISCRETE VARIABLES    615

GENERATION TIME      =        0.078 SECONDS    2.4 Mb      WIN200-121
EXECUTION TIME       =        0.218 SECONDS    2.4 Mb      WIN200-121

               S O L V E      S U M M A R Y
     MODEL   FRAUD               OBJECTIVE  Z
     TYPE    MIP                 DIRECTION  MAXIMIZE
     SOLVER  OSL                 FROM LINE  18470

**** SOLVER STATUS     2 ITERATION INTERRUPT
**** MODEL STATUS      8 INTEGER SOLUTION
**** OBJECTIVE VALUE              382.0000

 RESOURCE USAGE, LIMIT        333.891       750.000
 ITERATION COUNT, LIMIT    500078        500000

 OSL Version 1 Mar 21, 2001 WIN.OS.O2 20.0 058.043.039.WAT
 Work space allocated           --    3.65 Mb

 Relaxed optimum objective value:     507.20625
 Bound on best integer solution:      505.68750
 Objective value of this solution:    382.00000

 Relative gap: .24459 Absolute gap:   123.68750
 Optcr       : .10000 Optca:          0.0

 The solution does not satisfy the termination tolerances

**** REPORT SUMMARY :        0     NONOPT
                             0 INFEASIBLE
                             0  UNBOUNDED

E x e c u t i o n
----  18471 PARAMETER enssize              =         15.0

----  18472 VARIABLE  SELECT.L

m002 1.0,   m003 1.0,   m006 1.0,   m029 1.0,   m030 1.0,   m031 1.0
m032 1.0,   m033 1.0,   m034 1.0,   m036 1.0,   m037 1.0,   m039 1.0
m043 1.0,   m049 1.0,   m051 1.0
```

```
 LOOPS                            FOR/WHILE   8

MODEL STATISTICS
BLOCKS OF EQUATIONS       3     SINGLE EQUATIONS      565
BLOCKS OF VARIABLES       3     SINGLE VARIABLES      616
NON ZERO ELEMENTS     16727     DISCRETE VARIABLES    615

GENERATION TIME      =        0.078 SECONDS    2.4 Mb      WIN200-121
EXECUTION TIME       =        0.281 SECONDS    2.4 Mb      WIN200-121

               S O L V E      S U M M A R Y
     MODEL   FRAUD               OBJECTIVE  Z
     TYPE    MIP                 DIRECTION  MAXIMIZE
     SOLVER  OSL                 FROM LINE  18470

**** SOLVER STATUS     2 ITERATION INTERRUPT
**** MODEL STATUS      8 INTEGER SOLUTION
**** OBJECTIVE VALUE              377.0000

 RESOURCE USAGE, LIMIT        326.520       750.000
 ITERATION COUNT, LIMIT    500084        500000

 OSL Version 1 Mar 21, 2001 WIN.OS.O2 20.0 058.043.039.WAT
 Work space allocated           --    3.65 Mb

 Relaxed optimum objective value:     503.52849
 Bound on best integer solution:      502.33333
 Objective value of this solution:    377.00000

 Relative gap: .24950 Absolute gap:   125.33333
 Optcr       : .10000 Optca:          0.0

 The solution does not satisfy the termination tolerances

**** REPORT SUMMARY :        0     NONOPT
                             0 INFEASIBLE
                             0  UNBOUNDED

E x e c u t i o n
----  18471 PARAMETER enssize              =         17.0

----  18472 VARIABLE  SELECT.L

m002 1.0,   m003 1.0,   m019 1.0,   m029 1.0,   m030 1.0,   m031 1.0
m032 1.0,   m033 1.0,   m034 1.0,   m036 1.0,   m037 1.0,   m039 1.0
m043 1.0,   m047 1.0,   m048 1.0,   m049 1.0,   m052 1.0
```

```
LOOPS                            FOR/WHILE   9

MODEL STATISTICS
BLOCKS OF EQUATIONS       3     SINGLE EQUATIONS      565
BLOCKS OF VARIABLES       3     SINGLE VARIABLES      616
NON ZERO ELEMENTS     16727     DISCRETE VARIABLES    615

GENERATION TIME      =        0.062 SECONDS    2.4 Mb      WIN200-121
EXECUTION TIME       =        0.265 SECONDS    2.4 Mb      WIN200-121

               S O L V E      S U M M A R Y
     MODEL   FRAUD               OBJECTIVE  Z
     TYPE    MIP                 DIRECTION  MAXIMIZE
     SOLVER  OSL                 FROM LINE  18470

**** SOLVER STATUS     2 ITERATION INTERRUPT
**** MODEL STATUS      8 INTEGER SOLUTION
**** OBJECTIVE VALUE              373.0000

 RESOURCE USAGE, LIMIT        327.221       750.000
 ITERATION COUNT, LIMIT    500036        500000

 OSL Version 1 Mar 21, 2001 WIN.OS.O2 20.0 058.043.039.WAT
 Work space allocated           --    3.65 Mb

 Relaxed optimum objective value:     498.56604
 Bound on best integer solution:      497.49512
 Objective value of this solution:    373.00000

 Relative gap: .25024 Absolute gap:   124.49512
 Optcr       : .10000 Optca:          0.0

 The solution does not satisfy the termination tolerances

**** REPORT SUMMARY :        0     NONOPT
                             0 INFEASIBLE
                             0  UNBOUNDED

E x e c u t i o n
----  18471 PARAMETER enssize              =         19.0

----  18472 VARIABLE  SELECT.L

m002 1.0,   m005 1.0,   m015 1.0,   m017 1.0,   m019 1.0,   m021 1.0
m029 1.0,   m030 1.0,   m031 1.0,   m032 1.0,   m033 1.0,   m034 1.0
m035 1.0,   m036 1.0,   m037 1.0,   m039 1.0,   m043 1.0,   m047 1.0
m051 1.0
```

```
LOOPS                            FOR/WHILE   10

MODEL STATISTICS
BLOCKS OF EQUATIONS       3     SINGLE EQUATIONS      565
BLOCKS OF VARIABLES       3     SINGLE VARIABLES      616
NON ZERO ELEMENTS     16727     DISCRETE VARIABLES    615

GENERATION TIME      =        0.078 SECONDS    2.4 Mb      WIN200-121
EXECUTION TIME       =        0.235 SECONDS    2.4 Mb      WIN200-121

               S O L V E      S U M M A R Y
     MODEL   FRAUD               OBJECTIVE  Z
     TYPE    MIP                 DIRECTION  MAXIMIZE
     SOLVER  OSL                 FROM LINE  18470

**** SOLVER STATUS     2 ITERATION INTERRUPT
**** MODEL STATUS      8 INTEGER SOLUTION
**** OBJECTIVE VALUE              372.0000

 RESOURCE USAGE, LIMIT        375.900       750.000
 ITERATION COUNT, LIMIT    500001        500000

 OSL Version 1 Mar 21, 2001 WIN.OS.O2 20.0 058.043.039.WAT
 Work space allocated           --    3.65 Mb

 Relaxed optimum objective value:     491.86096
 Bound on best integer solution:      490.95455
 Objective value of this solution:    372.00000

 Relative gap: .24229 Absolute gap:   118.95455
 Optcr       : .10000 Optca:          0.0

 The solution does not satisfy the termination tolerances

**** REPORT SUMMARY :        0     NONOPT
                             0 INFEASIBLE
                             0  UNBOUNDED

E x e c u t i o n
----  18471 PARAMETER enssize              =         21.0

----  18472 VARIABLE  SELECT.L

m003 1.0,   m007 1.0,   m015 1.0,   m017 1.0,   m021 1.0,   m029 1.0
m030 1.0,   m031 1.0,   m032 1.0,   m033 1.0,   m034 1.0,   m035 1.0
m036 1.0,   m037 1.0,   m042 1.0,   m043 1.0,   m046 1.0,   m047 1.0
m049 1.0,   m050 1.0,   m051 1.0

USER: Course License                                G020111:1129AP-WIN
      Naval Postgraduate School, Operations Research             DC1696

**** FILE SUMMARY

INPUT      C:\WINDOWS\GAMSDIR\FRAUD3.GMS
OUTPUT     C:\WINDOWS\GAMSDIR\FRAUD3.LST
```

LIST OF REFERENCES

1 Defense Finance and Accounting Service, *Improper Payments/Data Mining Support: Final Report 831-583-3002*, Federal Data Corporation, Contract N00244-96-D-8055, 1999

2 Jones-Oxendine, Shawn R., *An Analysis of DOD Fraudulent Vendor Payments*, Master's Thesis, Naval Postgraduate School, Monterey, California, September 1999

3 Brief at Naval Postgraduate School, *The State of DFAS*, Mark A. Young, Rear Admiral, SC, USN, February 2000

4 DFAS IR Seaside to DFAS Headquarters Memorandum, Second Quarter Progress Report, March 2002

5 American Institute of Certified Public Accountants, Inc., *Consideration of Fraud in a Financial Statement Audit: Statement on Auditing Standards No. 82*, AICPA, New York, N.Y., 1997

6 American Institute of Certified Public Accountants, Inc., *Performance of Work, Audit Considerations for Irregularities: Statement on Auditing Standards No. 8*, AICPA, New York, N.Y., 1988

7 American Institute of Certified Public Accountants, Inc., *Analytical Procedures: Statement on Auditing Standards No. 56*, AICPA, New York, N.Y. 1988

8 Wheeler, Stephen and Pany, Kurt, "Assessing the Performance of Analytical Procedures: A Best Case Scenario," *The Accounting Review*, Vol. 65, No. 3, pp 557-577, July 1990

9 Busta, Bruce and Weinberg, Randy, "Using Benford's Law and Neural Networks as a Review Procedure," *Managerial Auditing Journal*, Vol. 13, No 6, pp 356-366, 1998

10 Davis, Jefferson T.; Massey, Anne P.; and Lovell II, Ronald E.R., "Supporting a Complex Audit Judgement Task: An Expert Network Approach," *European Journal of Operational Research*, Vol. 103, pp 350-372, December 1997

11 U.S. General Accounting Office, GAO Report GAO-02-69G, *Strategies to Manage Improper Payments: Learning from Public and Private Sector Organizations*, US General Accounting Office, Washington D.C., October 2001

12 U.S. General Accounting Office, GAO Report, GAO-01-44, *Financial Management: Billions in Improper Payments Continue to Require Attention*, US General Accounting Office, Washington D.C., October 2000

13 *Clementine 6.0 Users Manual*, SPSS Inc, Chicago, IL, 2001

14 Email communication between Dean Abbott and the author, Monterey, California, 22 Feb 2002

15 Dixon, Matt, *Ensembles and Their Applications*, University of Montana In-House Project, DTIC ADA 20010307 163, December 2000

16 Davia, Howard, *Fraud 101*, John Wiley and Sons, Inc., New York, NY, 2000

17 Crowder, Nita, "Fraud Detection Techniques," *Internal Auditor*, Vol. 54, Issue 2, pp 17-20, April 1997

18 Calderon, Thomas G. and Green, Brian P., "Internal Fraud Leaves Its Mark: Here's How to Spot, Trace and Prevent It," *The National Public Accountant*, Vol. 39, Issue 8, pp 17-19,36-38, August 1994

19 Calderon Thomas G. and Green, Brian P., "Signalling Fraud by Using Analytical Procedures," *The Ohio CPA Journal*, Vol. 53, Issue 2, pp 27-38, April 1994

20 Conover, W.J., *Practical Nonparametric Statistics*, 3rd Edition, John Wiley and Sons, Inc., New York, N.Y., 1999

21 Hand, D.J., *Construction and Assessment of Classification Rules*, John Wiley and Sons, Inc., New York, N.Y. 1997

22 Hand, David; Mannila, Heikki; and Smyth, Pedhraic, *Principles of Data Mining,* MIT Press, Cambridge, Massachusetts, 2001

23 Wilson, Arlette C. and Colbert, Janet, “An Analysis of Simple and Rigorous Decision Models as Analytical Procedures,” *Accounting Horizons*, Vol. 3, Issue 4, pp 79-83, December 1989

24 Brooke, Anthony; Kendrick, David; Meeraus, Alexander; Raman, Ramesh, *GAMS: A Users Guide*, GAMS Development Corporation, Washington D.C., December 1998

25 *S-PLUS 2000 Users Guide*, MathSoft Inc., Seattle, Washington, May 1999

26 W.N. Venables and B.D. Ripley, *Modern Applied Statistics with S-PLUS*, Third Edition, Springer, New York, N.Y. 1999

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Professor Samuel E. Buttrey.....1
   Code OR/SB
   Naval Postgraduate School
   Monterey, California 93943

4. Professor Lyn R. Whitaker.....1
   Code OR/LW
   Naval Postgraduate School
   Monterey, California 93943

5. Defense Finance and Accounting Service.....1
   Attn: Dave Riney
   Internal Review
   400 Gigling Rd
   Seaside, California 93955

6. LT Donald J. Jenkins.....1
   702 Atkins St
   Middletown, CT 06457

THIS PAGE INTENTIONALLY LEFT BLANK