# Improper Payment Detection in Department of Defense Financial Transactions[1]


Dean Abbott
Abbott Consulting
San Diego, CA
dean@abbott-consulting.com

Haleh Vafaie, PhD.
Federal Data Corporation
Bethesda, MD
Hvafaie@feddata.com

Mark Hutchins
Federal Data Corporation
Bethesda, MD
mhutchins@feddata.com

David Riney
Defense Finance and Accounting Service,
Seaside, CA
david.riney@dfas.mil


## ABSTRACT


This paper describes ongoing work to aid examiners in the detection of illegal, improper, or unusual transactions conducted against the Department of Defense (DoD) financial assets. The main goals of this project are to improve the Defense Finance and Accounting Service's (DFAS) ability to detect these payments and to reduce the manpower required to research them. By taking advantage of current data mining tools, DFAS hopes to enhance the precision of their predictions.

The challenges of using data mining to address this business problem included establishing a data set of known fraudulent payments, a target population of normal payments, and a method by which to leverage the known cases in the training of detection models. As is typical in fraud detection, the set of known cases was very small relative to the number of non-fraud examples. The authors used a cross-validation approach so that all the known fraud payments could be used in training. Then after generating more than 90 models, eleven models were selected, and their individual decisions were combined via voting to make the final decision. The results improved sensitivity to the known fraud payments significantly while keeping false alarms acceptably low.


## 1. INTRODUCTION

Detecting fraud is a constant challenge for any business. Both the private and public sectors have worked extensively to detect fraud in their systems. The extent of fraud, while varying from application to application, can be quite large. Cellular phone fraud alone costs the industry hundreds of millions of dollars per year (Fawcett and Provost, 1997), and fraudulent transfers of illegally obtained funds (money laundering) are estimated to be as much as $300 billion annually (Jensen, 1997).

The Defense Finance and Accounting Service (DFAS) is responsible for disbursing nearly all of the Department of Defense (DoD) funds. In its effort to be outstanding stewards of these funds, DFAS is determined to minimize fraud against DoD financial assets and has chosen Data Mining as one of its strategies to detect, and ultimately to deter fraud.

The first phase of this project involved selecting a data mining tool that best met our requirements. After reviewing more than thirty data mining tools on the market, the five best were thoroughly evaluated, and SPSS Inc.'s Clementine data mining product was acquired for this project. A description of the process used to select the data mining tool is outlined in Abbott, Matkovsky, and Elder (1998).

The challenges of using data mining to address this business problem included establishing a data set of known fraudulent payments, a target population of non-fraud, and a method by which to leverage the known fraud cases in the training of detection models. As is typical in fraud detection, the set of known


[1] Presented at the Federal Data Mining Symposium , Washington, DC, March 28-29, 2000.

cases was very small relative to the number of non-fraud examples. Thus, the researchers had to devise methods to reduce false alarms without drastically compromising the sensitivity of trained models. Although the approach outlined below was devised for a particular application, the principles undergirding the method can be applied across a wide range of applications.

## 2. DATA MINING APPROACH

### Obtaining the Data

DFAS decided to apply the data mining process to payment systems where millions of vendor payments are made annually using three different vendor pay systems located at about twenty sites. DFAS decided to limit the initial data mining effort to four sites that represent two of the payment systems, allowing 1.4 million payments to be reviewed.

Until DFAS consolidates its multiple vendor pay systems into a single pay system, the Defense Manpower Data Center (DMDC) receives an extract from each system then stores them in a Common Data Format (CDF) to standardize many of the data elements. Although this process alleviated the initial problem of standardization, the problem of what data to use in the extracts still existed between the different systems. Because different elements are populated in each of the systems, researchers began a complicated process of deciding which data transformations could be made to facilitate the data mining.

For the data mining process, DFAS extracted actual transactions for some of the cases and used source documents to recreate the remaining transactions to appear as they would have in the CDF format. The result was a data set of fraudulent payment candidates that DFAS used to develop models predicting similar transactions. Because many of the transactions were older than our data history, some data "reconstructions" were incomplete. For example, some transactions were missing data on payment type or payment method. Fortunately for the taxpayer, the number of known fraudulent transactions is small. However, this was the challenge for our Data Mining effort: trying to predict suspicious payments using a very small set of known fraudulent payments relative to a larger population of non-fraudulent payments.

### Transforming the Data

Initially, an effort was made to identify transformations common to both the known fraudulent payment data and the CDF data. Experts in identifying vendor payment fraud hypothesized dozens of potentially useful transformations of known information that might be useful indicators of fraud. In addition, transformations that required information not available in the known fraudulent payment data, but available in the CDF data, were included in the data set to be used in future unsupervised learning modeling stages of the project.

Examples of data transformations made in the first phase included setting flags that identified:

- Payments addressed to P.O. Box or Suite;
- Invoices from the same vendor paid to multiple addresses;
- Invoices from multiple vendors paid to the same address;
- Invoices from the same vendor were not sequential based on date submitted.

Rather than using a single fraud/not-fraud binary label for the output variable, four fraudulent payment types, called types A, B, C, and D, were identified as comprising the different styles of payments in the known fraud data. Although separating the known fraudulent payments into types was not essential to the modeling process, researchers did believe that the additional information would aid classifiers in identifying suspicious payments, and reduce false alarms. The primary reason for this belief was that the types of known fraudulent payments had significantly different behaviors, which would cause a classification algorithm to try to segment the fraudulent payment data into groups anyway. By creating the payment types beforehand, simpler models were more likely to be formed.

## Creating Training, Testing, and Validation Subsets

Avoiding Overfit. A procedure of using training, testing, and validation data sets was used throughout the model building process to reduce the likelihood of overfitting models. Overfitting has several harmful side effects, first of which is poorer performance on unseen data than is expected from the training error rates (Jensen, 1999). Additionally, it also has the effect of selecting a model that includes extra variables or weights that are spurious. Therefore, not only is the model less efficient, but it also identifies the **wrong** patterns in the data. If our objective is to find the best model that truly represents the patterns in the data, great care should be taken to avoid overfit.

The common practice of using both a training data set to build a model and a testing data set to assess the model accuracy reduces the likelihood of overfit. However, the data mining process is usually iterative, including the training/testing cycle. The testing data set provides an indication of how well the model will perform on unseen data. But after several iterations of training and testing, the testing data set ultimately guides the selection of candidate inputs, algorithm parameters, and "good" models. Therefore, the testing data set ceases to be independent and becomes an integral part of the model itself. To guard against this effect, a third data set, called the *validation* data set, is used to assess model quality at the end of the training/testing cycles. The validation data set is used only once, so that it remains independent of any model input or parameter selection and can be safely used to select models.

Before dividing the data into training, testing, and validation sets, there were three problems to overcome specific to the data set used in this application.

Data Problem 1: Small numbers of labeled fraud payments. The first difficulty, common in fraud detection problems, was the small number of labeled fraud payments or transactions available for modeling. The difficulty was particularly acute here with our desire to have three data subsets for each model (training, testing, and validation). However, with precious few known fraudulent payments available, keeping any of them out of training data meant that patterns of fraudulent payment behavior *may* have been withheld from the models, and therefore missed once the models were deployed. The question therefore was "how does one make best use of the limited data available, and capture *every* pattern that exists in the known fraudulent payment data?"

Cross-validation is the methodology used typically to overcome this data problem. In cross-validation, data are split into training, testing, and sometimes validation data sets multiple times, so that each fraud payment is included in each of the data sets. The number of payments used in the training set can include all the payments except one, keeping the unused payment for testing, and repeat the splitting until each payment has been excluded from the training set (included in the testing set) one time. However, because thousands of payments were used in training our models, this was deemed too computationally expensive. Most of the benefit of cross-validation can be achieved from a much smaller number of cross-validation folds, and for this project, it was decided that 11 subsets would be sufficient.

Data Problem 2: Payments are not independent. A second problem related to the assumption of independence of database rows—individual payments—to one another. Typically, the splitting of data into training, testing, and validation subsets, is done randomly. However, this procedure assumes independence between rows (payments) in the data. However, upon further examination, it became clear that the modeling variables that were a part of the payments tended to be correlated within each payee, because payees tend to invoice for work that is similar from payment to payment. For example, a telephone company may have a similarly sized invoice for long distance service each month. If some payments from a payee are included in both the training and testing subsets, testing results will be unrealistically optimistic. This occurs because the same pattern or a very similar pattern will exist for the same payee in both the training data (which is used to create a model) and the testing data (which is used to assess the model). When the model is deployed, most of the payees will not have been seen during training; the purpose of the model is to generalize payment patterns, not profile individual payees.

The solution was not only to select payees randomly and to include each in only one of the data sets (training, testing, *or* validation), but also to keep together all the fraudulent payments associated with each payee during the split. For example, if XYZ Corporation was chosen to be included in the testing data, all other payments made to XYZ Corporation were also included in the testing data. However, to avoid biasing the data too heavily towards a payee that has many payments in the system (perhaps several hundred), a limit of 100 payments per payee was specified.

Data Problem 3: Large numbers of unlabelled non-fraudulent payments. A third problem resulted from the lack of adjudicated non-fraudulent payments in the general population. These payments may have been improper or non-fraudulent, and this label was unknown when collected in the database. It was assumed for the purposes of modeling, however, that they were non-fraudulent because this was far more likely to be true. Additionally, for small data sets, it was unlikely that any of the payments would have been labeled incorrectly as non-fraudulent when they were actually fraudulent.

However, for larger data sets, perhaps over 100,000 payments, the likelihood of having mislabeled payments included in the training or testing sets was much higher. The modeling classifiers could become confused if too many mislabeled payments are included. If the mislabeled payments happened to correspond to patterns of fraud, these payments may be dismissed erroneously in an effort to reduce false alarms.

To compensate for this effect, the training data set was kept relatively small compared to the full 1.4 million-payment data set. It was determined that training data sets of approximately 4,000 payments each would keep the likelihood of mislabeled payments sufficiently small so that training would not be negatively effected. The testing data sets were selected to have 2,000 payments from the population. By keeping the data sets small, the time needed to train and test models was also reduced significantly. A much larger 125,000-payment data set was used for model validation.

Splitting the Data into Subsets. Payments from the known fraud data and the general (non-fraudulent payment) population were split into the 11 training, testing, and validation subsets separately, and joined together after the splits were performed. Additional considerations were given to the non-fraudulent payment population for generating diversity, but will not be discussed in depth here. In brief, one fourth of the payments in each subset (1,000 for training, 500 for testing) was chosen by selecting a *payees* randomly and including all other payments with the same payee (until 1,000 payments were selected for training data, 500 for testing data). The second quarter of the payments in each subset was chosen by selecting a *contract* randomly and including all other payments with the same contract number (without replacement of payments already selected). The third quarter was chosen by selecting an *address* randomly, and including all payments with that same address, and the final quarter included payments selected at random. For each payee, contract, or address sample, a maximum of 100 payments were included in the sample from the general population to limit the influence of any single payee, contract, or address on the models. This process was performed 11 times, once for each split, so that a total of approximately 44,000 total (distinct) training samples and 22,000 (distinct) testing samples were drawn from the 1.4 million -payment population.

For validation data, a much larger set was used: 125,000 payments, but rather than repeating the selection of payments for each split, the same 125,000 were used for all splits. Additionally, to prevent bias toward any payee, contract, or address, the entire sample was selected randomly from the 1.4 million payments (minus those payments used for training and testing). This procedure for validation data had the additional benefit of providing better comparisons for models generated from data in each of the 11 splits; they were all validated with the same non-fraudulent payment data.

Known fraud data, with the four type labels (A, B, C, and D), were split following a different procedure. For types A, B, and C, all payments associated with one payee were selected randomly for testing data, all payments associated with a random payee were selected for validation data, and all the payments associated with the remaining payees were put into the training data. In this way, we included as many patterns in the training data as possible for each split. For type D, because so few payees were available, the payments were split randomly between training and testing data sets, and another payee was held out for validation. Then known fraud splits were combined with the non-fraudulent payment data in each of the 11 splits. We

must note that the validation data set was the same for each of the 11 splits, while the fraudulent payment data sets were not.

After the data splits and combinations, there were 11 training data sets containing approximately 4,000 payments each, and 11 testing data sets containing approximately 2,000 payments each. For the validation data, it was determined that the same 125,000 non-fraudulent payments would be used in each of the 11 validation data sets. The known fraud data reserved for the validation sets contained payments not included in the respective training or testing sets.

## Procedure for Creating Models

A template Clementine stream containing nodes for reading data, assigning data types, performing simple data transformations, and displaying model results was available for all modelers. Figure 1 below shows the template stream.

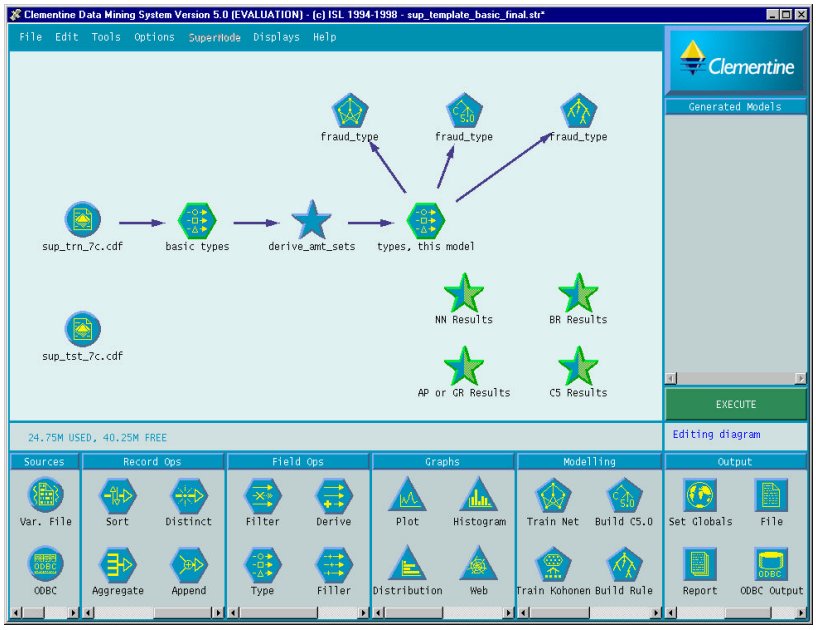


**Figure 1: Clementine Stream**

Ten modelers produced candidate models, with each modeler responsible for a subset of the data splits (typically 2 splits per modeler). However, each modeler was required to create candidate models using at least two of the algorithms available for use in Clementine: decision trees, rule induction, neural networks, and association rules. Modelers were encouraged to use a variety of algorithm parameter settings as well. The diversity of final models, generated from varied data sets (each with a different subset of fraudulent payment data), modelers, and algorithms, were all important to the selection of models to include in a final model ensemble.

Over 90 models were generated at the end of the modeling phase. At this stage, we decided to include several of the developed models in the final system in order to reduce the variance and bias of the final selection of results on the large unseen data set. As mentioned above, each model trained on a small sample split, which could result in a higher variance on the results of unseen data (Freidman, 1997). Yet by combining several models, a larger sample size or "view" of the data is used (in this case up to eleven different samples), and this should reduce the variance of the results. Additionally, it has been demonstrated that bias is reduced by combining models, also called creating model ensembles (Diettereich, 1997) (Abbott, 1999) (Elder and Ridgeway, 1999). The simplest form of model combining is a simple majority vote of the outputs. For the results obtained in this project, a payment is flagged as suspicious if

six or more of the final eleven models deem the payment anomalous, regardless of the type of payment. The next section describes the strategy we used to select these models.

### Criteria for Scoring Models

Because of multiple modelers, a standardized method of documentation was established so that all models could be evaluated equally. Within Clementine, models were set up to produce the total number of:

1) Payments labeled as fraudulent and correctly called fraud;
2) Payments labeled as Non-fraudulent and correctly called non-fraud;
3) Payments labeled as fraudulent and incorrectly called non-fraud;
4) Payments labeled as Non-fraudulent and incorrectly called fraud.

These outputs were recorded in a spreadsheet and were used to determine model performance. From these numbers, we calculated percentages for the following three areas:

1) Fraudulent payment Sensitivity: the percentage of fraudulent payments correctly called fraud from the known population. Sensitivity predicts the likelihood that we will find fraudulent payments in the general population. *(# Fraudulent payments called fraudulent / # fraudulent payments in data);*
2) False Alarm Rate: the percentage of non-fraudulent payments incorrectly called fraudulent from the general population.
3) True Alarm Rate: the percentage of fraudulent payments predicted as fraudulent from the entire population of payments predicted as fraudulent. True alarm rate predicts the density of fraudulent payments that will populate the list of fraudulent payment predictions. *(# Fraudulent payments called fraudulent that really were fraudulent / Total # payments called fraudulent).*

### Assessing Model Performance: Testing and Validation

Model evaluation used different weighting systems in order to emphasize the calculated percentages for both testing and validation results, especially for fraudulent payment sensitivity and false alarm rate. For example in one evaluation, we weighted testing and validation results equally. Yet in a second assessment, we gave more weight to the validation results. Scores for testing and validation results were then added together to produce an overall score and the overall score of a model was then ranked.

For the purposes of majority voting, we decided to choose eleven final models, which were chosen based on the following criteria:

1) Overall performance of model: Determined by a model's weighted and ranked score.
2) Algorithm Diversity: For example, we did not want all the final models to be neural networks.
3) Sample Split Representation (Modeler Diversity): To decrease variation and bias of results, we made sure that as many of the samples and modelers were represented as possible.

Any number of votes could be required to call a payment suspicious. A large number of votes (10 of 11, for example) would reduce false alarms at the expense of correctly catching fewer suspicious payments, while a small number of votes (1 or 2, for example), would produce too many false alarms. After a brief investigation, it was determined that a majority vote (six of eleven) provided the best tradeoff.

## 3. MODELING RESULTS

In order to validate our strategy and methodology, we included the eleven best models in the final decision making process. Our ultimate goal was to minimize the false alarms (non-fraudulent payments that were flagged by the models) and maximize sensitivity (known fraudulent payments that were flagged by the models). As described above, the final decision was based on the majority vote; if a majority of the final models classified a payment as "suspicious," that payment was labeled an "anomaly" and a candidate for further investigation. With 11 total models, a majority vote meant that six or more models must flag the

payment. Table 1 lists the algorithms used in the final combination: five neural networks, four decision trees, and two rule sets.

**Tab1e 1: Algorithms Used to Create Models Included in Final Combination.**

| Model Number | Algorithm Type |
|---|---|
| 1 | Neural Network |
| 2 | Decision tree |
| 3 | Neural Network |
| 4 | Decision tree |
| 5 | Decision tree |
| 6 | Decision tree |
| 7 | Rule based |
| 8 | Neural Network |
| 9 | Neural Network |
| 10 | Neural Network |
| 11 | Rule based |

Figure 2 shows the model sensitivity of all 11 models and the combination on the validation data set, on a 1-5 scale, where a score of 5 was given to the most sensitive model, and 1 to the least sensitive. Note that the combination was much better than average (which was 2.9), and nearly the best model overall (model 10 had slightly higher sensitivity). This behavior is typical in voting combinations (Abbott, 1999).

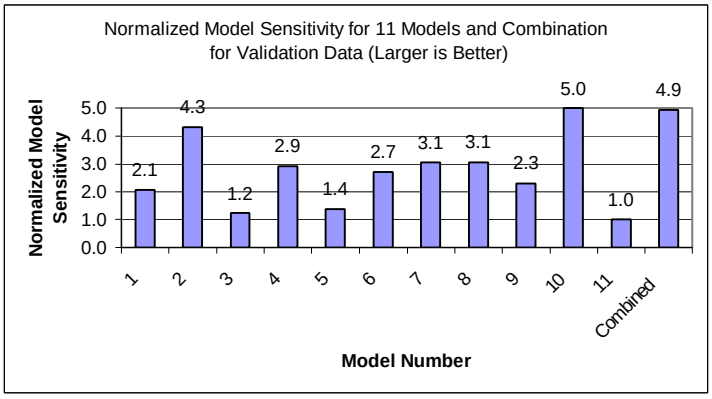


**Figure 2: Model Sensitivity Comparison**

In Figure 3, the false alarm performance is shown on a normalized scale ( with the lowest number of false alarms given a score of 1, and the most a score of 5). As with sensitivity, the combination model had a much lower score than the average model (which was 1.4), though model 6 had the overall lowest false alarm rate. It is clear, however, that *any* positive, non-zero weighting of sensitivity and false alarms results in the combination having the best score overall. For example, Figure 4 shows an equal weighting between sensitivity and false alarms. The false alarm contribution is reversed from 1 (best) to 5 (worst), to the range 5 (best) to 1 (worst) so that higher scores are better, lower worse. On average it is 32% better than single models, and is 7% better than its nearest competitor (model 10). It is clearly better to use the combination rather than trying to determine *a priori* which of the individual models to select.

Neural networks and decision trees performed nearly the same in sensitivity and false alarms on average. The rule sets lagged behind somewhat, though that was due to poor performance with model 11; the other rule set, model 7, performed nearly the same as the average neural network or tree. A neural network (model 10) had the best sensitivity overall, while a decision tree had the lowest false alarm (model 6). A summary is shown in Table 2 below. Therefore, no algorithm had a distinct advantage, and all seem to contribute to the success of the final model combination.

The results described here validate our hypothesis that by combining models we improve the overall performance of the system. It appears that just as with human decision making, the inclusion of multiple

"advisors" in the final model has the effect of reducing the risk in the decision and improving overall performance.

**Table 2: Algorithm Performance Summary**

| Algorithm | Average Sensitivity | Average False Alarm | Average Total Score |
|---|---|---|---|
| Neural Network | 2.9 | 1.1 | 6.9 |
| Decision Tree | 3.0 | 1.0 | 7.0 |
| Rule Induction | 2.3 | 3.1 | 4.2 |
| Overall | 2.9 | 1.4 | 6.4 |

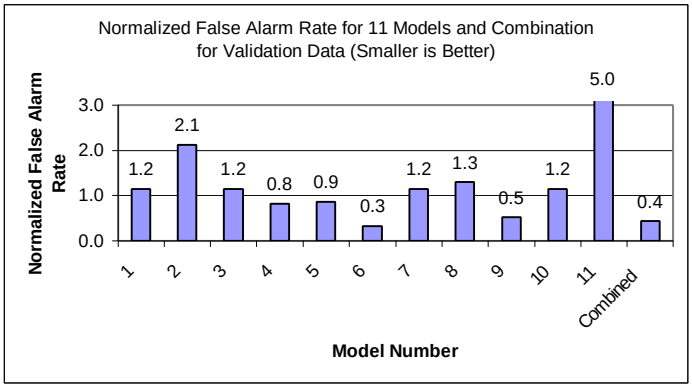


**Figure 3: False Alarm Rate Comparison**

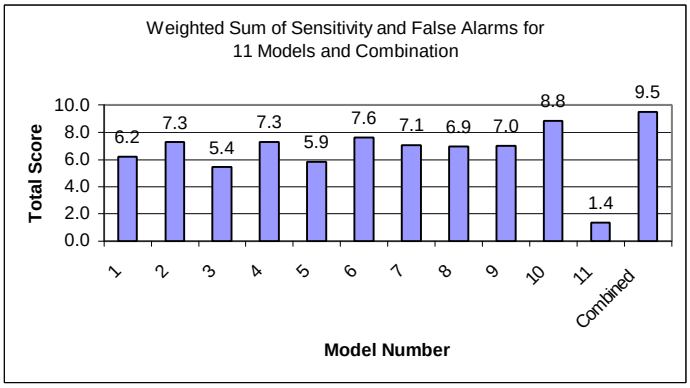


**Figure 4: Total Weighted Score Comparison**

## 4. DISCUSSION

A procedure has been outlined for overcoming problems common in fraud detection applications, including paucity of known fraud data, large amounts of unlabeled data which is nearly always non-fraudulent, and using as much of the data as possible so that all of the known patterns of fraud are captured. The procedure required paying careful attention to subtle database issues, such as the dependence of payments in the data, before splitting the data into training, testing, and validation data sets. The results also show the benefit of model ensembles to reduce the risks associated with the selection of a single model, to improve model sensitivity and to reduce false alarms.

It is unclear however, as to which of the factors in the data design and modeling stages contributed the most to the final favorable results, or how many models were necessary to obtain the benefits of combining model outputs. Answers to the questions have practical significance and should be explored in future work.

## 5. ACKNOWLEDGEMENTS

This work was contracted by DFAS with EDS as the prime contractor, Federal Data Corporation, Elder Research, and Abbott Consulting as subcontractors. The authors wish to thank members of the DFAS team, (Lance Wright, Karen Forbes, Randal T. Faulkner, LTC. Chris Drews, and CDR. Kevin Hale), and the DMDC team (Margot Wolcott and Jack Maroney) for their insightful comments and creative modeling throughout the project. Additionally, we thank Philip Matkovsky of Federal Data Corporation and Dr. John F. Elder of Elder Research for their helpful critiques and technical input throughout the project.

## 6. REFERENCES

Abbott, D.W., I.P. Matkovsky, J.F. Elder (1998). An Evaluation of High-end Data Mining Tools for Fraud Detection. *1998 IEEE Int'l Conference on Systems, Man, and Cybernetics*, San Diego, CA, October 12-14.

Abbott, D.W. (1999). Combining Models to Improve Classifier Accuracy and Robustness. *1999 International Conference on Information Fusion—Fusion99*, Sunnyvale, CA, July 6-8.

Dietterich, T. (1997). Machine-Learning Research: Four Current Directions. *AI Magazine*, 18[4], 97-136.

Elder, John F. and Greg Ridgeway (1999). Combining Estimators to Improve Performance: Bundling, Bagging, Boosting, and Baysian Model Averaging. *International Conference on Knowledge Discovery and Data Mining*, San Diego, CA, August 15.

Fawcett, T., and F. Provost (1997). Adaptive Fraud Detection. *Data Mining and Knowledge Discovery*, Kluwer Academic Publishers, 1(3), 291-31.

Friedman, J. H. (1997). On Bias, Variance, 0/1—Loss, and the Curse-of-Dimensionality. *Data Mining and Knowledge Discovery*, Kluwer Academic Publishers, 1(1), 55–77.

Jensen, D., (1995). Prospective Assessment of AI Technologies for Fraud Detection: A Case Study. *Working Papers of the AAAI-97 Workshop on Artificial Intelligence Approaches to Fraud Detection and Risk Management*, July.

Jensen, D., and P.R. Cohen (2000). Multiple Comparisons in Induction Algorithms. *Machine Learning Journal*, 38(3), 1-30.

## 7. BIOGRAPHIES

Dean Abbott is the founder and president of Abbott Consulting which focuses on data mining, pattern recognition and knowledge discovery consulting services. Mr. Abbott has over a decade of experience as a data mining researcher solving technical problems by discovering patterns in data. Applications have included fraud detection, signal and image processing, guidance and control, direct marketing, and stock market timing. He has been an invited lecturer on data mining techniques and software products to both technical and non-technical audiences, and has been co-instructor of a well-regarded survey course on data mining.

Haleh Vafaie is a senior scientist and manager for the data mining group at Federal Data Corporation. Her research interests include data mining, data modeling, intelligent information extraction, knowledge acquisition, decision support systems, and genetic algorithms. Dr. Vafaie has over 13 years of specialized experience in data mining and machine learning and is the author or co-author of 25 publications in the area of machine learning. She received her Ph.D. in Information Technology and Engineering from George Mason University in 1997.

Mark Hutchins is a Senior Research Consultant and Project Manager for Federal Data Corporation's Analytical Systems Group. Mr. Hutchins works on data mining and other related projects and has 8 years

of experience in the specialized field of modeling and simulation, with emphasis on using higher-order statistical methods to develop an understanding of complex data sets.

David Riney is Branch Chief for Federal Payments with Operation Mongoose, a unit within the Defense Finance and Accounting Service. Mr. Riney is responsible for reviewing payments to Department of Defense Contractors for potential fraud. Mr. Riney has a BS in Accounting and a MS in Information Systems and has over 22 years of experience working with the Department of Defense.