Some Experimental Issues in Financial Fraud Mining

Article *in* Procedia Computer Science · December 2016

DOI: 10.1016/j.procs.2016.05.515

Some Experimental Issues in Financial Fraud Mining

Jarrod West[1] and Maumita Bhattacharya[2]
School of Computing and Mathematics
Charles Sturt University, NSW, Australia.
[1]jnwest@netspace.net.au, [2]mbhattacharya@csu.edu.au

Abstract
Financial fraud detection is an important problem with a number of design aspects to consider. Issues such as problem representation, choice of detection technique, feature selection, and performance analysis will all affect the perceived ability of solutions, so for auditors and researchers to be able to sufficiently detect financial fraud it is necessary that these issues be thoroughly explored. In this paper we will analyse some of the relevant experimental issues of fraud detection with a focus on credit card fraud. Observations will be made on issues that have been explored by prior researchers for general data mining problems but not yet thoroughly explored in the context of financial fraud detection, including problem representation, feature selection, and performance metrics. We further investigated some of these issues with controlled simulations, concentrating on detection algorithms, feature selection, and performance metrics for credit card fraud.

Keywords: Financial fraud detection; credit card fraud; data mining; computational intelligence

1 Introduction and Background

The increase in instances of financial fraud that has occurred in recent years is a serious problem with wide-spread ramifications (Bhattacharyya et al., 2011). As well as the obvious immediate consequences fraud can have long-term repercussions by reducing consumer confidence, supporting organised crime, and affecting people's cost of living (West and Bhattacharya, 2016a), (West, Bhattacharya and Islam, 2014). Examples of fraud include stolen credit cards and phishing scams for credit card fraud (Quah and Sriganesh, 2008), or earnings manipulations to improve the public appearance of a company for financial statement fraud (Ravisankar et al., 2011), (Zhou and Kapoor, 2011). Due to its impact on general society in both the short term and in the future thwarting financial fraud should be considered a highly important task for computer science researchers (Bhattacharyya et al., 2011).

Financial fraud detection aims to take large quantities of transactional data and separate the fraudulent samples from legitimate ones. Because of this it is inherently a data mining problem and as such it has the same fundamental requirements as similar tasks (Duman and Ozcelik, 2011). Understanding the problem domain, such as the vast imbalance between the number of fraudulent and legitimate transactions or the difference between the various types of fraud, is imperative to proceed correctly with a detection solution (Dal Pozzolo et al., 2014), (Duman and Ozcelik, 2011). Feature selection is necessary to identify which aspects of the transaction correlate with fraudulent behaviour and which don't. Selecting suboptimal features may both weaken the capabilities of the solution and increase the computation time (Kantardzic, 2011). When assessing the performance of a potential solution the metrics chosen need to correctly reflect the desired outcomes. For example, the fast-paced nature of some forms of fraud may mean that timeliness of results is more important than achieving the highest classification accuracy (Quah and Sriganesh, 2008).

Over the years various computational methods have been used for fraud detection and, like other similar problems, successful implementation of the detection methods depends on having a clear understanding of the problem domain. While some prior researchers have focussed on the common issues such as problem representation for data mining problems in general there has been almost no analysis from the perspective of fraud detection which we aim to address here.

Hall and Holmes performed a comparison of several universal feature selection methods for data mining, using software to rank their effectiveness against various datasets (Hall and Holmes, 2003). Yang and Wu conducted a survey of current data mining researchers to identify which issues were common across the breadth of the field (Yang and Wu, 2006). Quah and Sriganesh experimented on credit card fraud with a self-organising map, visualising the subsequent clusters to determine the results (Quah and Sriganesh, 2008). Sánchez et al. used numerical and textual details from the client and transaction, performing a detailed analysis of the dataset to determine which attributes were the most relevant to detecting fraud (Sánchez et al., 2009). Panigrahi et al. looked at different aspects of the problem and broke their solution down into four distinct steps: filtering, combination, classification, and learning (Panigrahi et al., 2009).

Ravisankar et al. focussed on financial statement fraud, comparing the performance of various data mining methods including neural networks, support vector machine, group method of data handing, logistic regression, and genetic programming (Ravisankar et al., 2011). Zhou and Kapoor looked at common behaviours that are frequently present for financial statement fraud and created a framework to be used for designing detection methods (Zhou and Kapoor, 2011). Duman and Ozcelik considered two distinct sub-categories of credit card fraud: large scale fraud committed by organised crime groups, and opportunistic fraud with lost or stolen cards (Duman and Ozcelik, 2011). Wong et al. used true and false positive rates as the measure of success for their investigation (Wong et al., 2012), while Sahin et al. included features from the credit card's prior transactions to enable identification of behavioural differences (Sahin et al., 2013). Olszewski performed several experiments using a self-organising map to compare credit card fraud, telecommunications fraud, and network intrusion (Olszewski, 2014). Halvaiee and Akbari looked at credit card fraud, altering their solution to improve the performance on common metrics such as accuracy, hit rate, and false positive rate (Halvaiee and Akbari, 2014).

The remainder of the article is structured as follows. The next section gives an overview of some of the relevant financial fraud detection experimental issues including problem representation, feature selection, and performance metrics. Section 3 details the simulations that we will be undertaking to demonstrate these issues including our scientific method and the various algorithms and metrics we will investigate. Section 4 provides an analysis of our simulation results and discussion on the experimental outcomes. Section 5 concludes the paper with a summary of our findings and suggestions for future direction. Note that the experimental issues covered in our research are applicable to all forms of financial fraud detection, though for experimental convenience we will be concentrating on credit card fraud specifically.

2 Summary of Some Experimental Issues

2.1 Problem Representation

To be capable of solving a complex problem like financial fraud detection it is important to first obtain a complete understanding of the problem domain (West and Bhattacharya, 2015a), (West and Bhattacharya, 2015b), (West and Bhattacharya, 2016b). Fortunately, there are a number of well-defined and understood models that are already being utilised for both fraud detection and similar problems such as network intrusion such as regression, classification, visualisation, clustering, and rule-based. Regression is a traditional statistical method that has been used extensively in data mining for many years. It aims to expose relationships between a dependent variable and a set of independent variables (Ngai et al., 2011). Classification is a data mining method that separates a list of unknown samples into one of several discrete classes (Ngai et al., 2011). Binary classification is a simplified case in which there exists only two possible categories (such as fraudulent and legitimate).

Visualisation refers to any data mining method that results in the presentation of data into a clear and understandable format for the purpose of being manually observed by a human operator. People are naturally adept at using patterns to understand complex problems, and exploiting this fact can be a powerful tool in comprehending the results of data mining problems (Kantardzic, 2011). Similar to classification, clustering is a method that is used to split samples into distinct, related groups that have no affiliation to other categories (Ngai et al., 2011). A clustering model makes use of a measure of similarity to assign input samples to clusters within a dimensional space: samples which are calculated to have a high similarity are naturally grouped together into the same cluster. Association rules offer a simple form of classification based on established mathematical logic statements. A model is created that takes a set of attributes and forms a prediction on the outcome. This model is the combination of multiple rules with an antecedent, based on the input parameters; and a consequent, an outcome based on the antecedent (Han and Kamber, 2012).

2.2 Feature Selection

Regardless of the data mining model chosen feature selection is an integral part of solving any problem. Each method relies on processing large quantities of data to detect obscured relationships and meanings, and therefore the variables selected for inclusion must be a good representation of the data as a whole. As an example we will consider credit card fraud, where all of the transaction, client, and account details are features that may be used in detection algorithms.

The aim of feature selection is to improve both the actual and computational performance of the solution, as well as providing a better understanding of the problem. To this end, algorithms are used to rank or choose which features are the most applicable to the current task. Feature ranking algorithms make use of an evaluation method to assign a rating to individual features based on attributes such as consistency, accuracy, and content, and choose a subset of these based on that ranking. When used correctly this subset should have comparable ability to the full set while being significantly smaller.

There are several potential issues with credit-card fraud feature selection. Firstly, the training sets utilised for credit card fraud detection experiments are typically obtained from real-world financial institutions’ databases. The separation of legitimate and fraudulent transactions is based on existing detection methods and customer reporting, and it is inherently possible that more subtle forms of fraud have escaped the notice of consumers and auditors. Additionally, there are several privacy concerns surrounding the use of genuine financial information for credit card fraud detection research. Many researchers were not able to identify the dataset they used, or even the institution it was obtained from or the features they selected (Sahin et al., 2013).

2.3 Performance Metrics

Measuring the success of computational intelligence algorithms (Bhattacharya, 2008), (Bhattacharya et al., 2016) is an important step in determining their suitability, especially for a problem such as financial fraud where minor improvements in performance can lead to large economic benefits. Performance can be measured in many different ways, such as absolute ability, performance relative to other factors, probability of success, and more. Table 1 provides a brief description of several common performance metrics and the problem representation that they apply to (Fawcett, 2006), (Han and Kamber, 2012), (Kantardzic, 2011).

Category	Metric	Description
Classification	Accuracy	Ratio of samples correctly classified to total samples
	Sensitivity	Ratio of positive samples correctly classified to total positive samples. Also known as recall, true positive hit rate, or hit rate
	Specificity	Ratio of negative samples correctly classified to total negative samples
	Precision	Ratio of positive samples correctly classified to total samples classified as positive
	False positive rate	The inverse of the true positive rate, given as 1-specificity
	F-measure	The harmonic mean of precision and recall (sensitivity). Also known as F-score or F.
	Fβ	A form of F-measure that applies a weighting of β to the precision and recall, where β is a positive, real number
	Cost minimisation	Measures the effectiveness of an algorithm by minimising the total misclassification cost relative to each type of error
Statistical	Z-score	Measures the rate of change in a variable, either independently, with respect to its historical values, or against a similar variable
	Sum of squared error	The difference between two sets of values, squared to separate out distinct clusters of values
Association rule	Support	The percentage of samples that contain a given itemset (group of items that commonly occur together in the problem space)
	Confidence	The proportion of samples that match a specific rule against the total that include the antecedent
	Lift	A correlation measure used to determine whether an association rule is useful to the problem
	Conviction	A measure of the inaccuracy of the rule, or the chance of the antecedent occurring without the consequent
Clustering	Hopkins statistic	A measure of the probability that a variable is randomly distributed within a space, used to determine whether a dataset contains significant clusters
Visual	ROC curve	Receiver operating characteristic curve, a two dimensional graph that provides an easily interpreted visualisation of the success of a binary classification method
	AUC	The area under an ROC curve, given between 0 and 1. Coalesces both the true and false positive rates into a single measurement

Table 1: Common performance metrics

3 Simulation Details

To investigate the ability of classification techniques, capacity of feature selection algorithms, and efficacy of performance metrics we have run a number of experiments using binary classification methods and analysed the results. In particular we looked at accuracy, sensitivity, specificity, precision, false positive rate, F-measure, and a common variant of Fβ, F2. The following sections provide details on the various tests undertaken.

The experiments were performed on the UCSD credit card dataset using Java implementations of several algorithms from different branches of computational intelligence and data mining. We compared the performance of genetic programming (GP), genetic algorithms (GA), ant colony optimisations (ACO), neural networks (NN), support vector machines (SVM), fuzzy logic (FL), decision trees (DT), functions (Fn), lazy evaluators (Lazy), and rule-based classifiers (Rule). Additionally, we investigated the affects that attribute selection had on fraud detection using seven feature selection methods (FS1-7).

The experiments were undertaken on a machine with a Quad-Core 3.06GHz processor and 12GB of RAM. To improve reliability we made use of 10-fold cross validation to reduce the chance of statistical errors. More detailed information on the experiments is provided below.

3.1 Algorithms

The detection and feature selection algorithms used in our experiments are provided by Tables 2 and 3 respectively.

Algorithm	Description
GP1	Hybrid of two genetic programming approaches where individuals can be comprised of multiple classification rules that must result in the same class.
GP2	Represents each rule with a single context free grammar comprised of logical and relational operators similar to GP1, but with a single rule assigned to each individual.
GA1	Incrementally learning algorithm that assesses attributes individually and orders them based on their relevance to the problem.
GA2	Extended version of GA1 that can use either the best rule as the basis of the next generation or the entire ruleset.
ACO	Model of an ant colony with rules representing the path that each ant will follow.
NN1	Incrementally created neural network that uses distance weighting to construct its hidden neurons.
NN2	Iterative network that represents its neurons as learning vectors.
SVM	Variant of support vector machine that parameterises the number of support vectors.
FL	Fuzzy rule learner based on RIPPER algorithm.
DT1	Random forest implementation that generates small, separate classification trees using random combinations of input variables for the nodes.
DT2	Decision tree method that uses a best-first expansion when classifying a sample.
Fn	Logistic regression function that employs the LogitBoost algorithm to create a base regression learner, with further classification is employed using a tree-style model
Lazy	Learning approach that applies weighting to classification instances resulting in non-linear estimations.
Rule	Simplistic method that uses basic classification rules to build a straightforward model.

Table 2: Detection algorithms used in the simulations

3.2 Dataset and Preprocessing

The dataset used in our research is a synthesised credit card dataset used for the 2009 UCSD-FICO data mining contest. It consists of entirely numerical data with 334 input attributes and 10000 records. Like real-world credit card problems the UCSD dataset has an extreme imbalance between instances of each class, legitimate samples outnumbering fraudulent ones in a ratio of 91:9.

The Ant Colony Optimisation implementation that we used for our experiments requires inputs to be entered in a non-continuous format. As such for experiments using this algorithm the dataset was first preprocessed with a Bayesian discretiser to convert each attribute to discrete values before conducting the main experiment. This discretiser uses an approximation of Bayes algorithm to create probability curves for each class. Intervals are created on the boundaries between curves such that each represents a discrete value that each attribute can be assigned to.

Algorithm	Description
FS1	Method that focusses on the ability of each feature to predict the resulting class as well as increasing correlation between attributes and avoiding redundancy
FS2	Approach that limits itself to correlation between attributes, utilising a weighted average for nominal features so they can also be studied on an individual basis.
FS3	Technique that utilises the gain ratio function to measure the information gained by each attribute and its effectiveness at predicting the class.
FS4	Uses the information gain the attribute provides for the resulting class.
FS5	Ranks the usefulness of attributes with the OneR classifying algorithm.
FS6	Method that considers the effectiveness of a feature by evaluating its ability to classify against similar instances of both classes.
FS7	Measures the value of a feature by calculating its symmetric uncertainty at evaluating a given class..

Table 3: Feature selection algorithms used in the simulations

4 Results and Analysis

Table 4 provides results for many of the binary classification metrics given in the previous section, and the percentage of classification error for each fold of the dataset is listed in Figure 1. Table 5 details the standard deviation of each combination of performance metric and detection algorithm, while Table 6 compares the percentage of feature similarity of each feature selection method's resulting dataset.

4.1 Detection Algorithms

Using our chosen performance metrics we can easily observe that many of the detection algorithms appeared to have impressive results on the dataset. Several of the computational intelligence algorithms such as GP1, GA1, GA2, FL, and SVM all had very low false positive rates of less than 0.002, as well as very high accuracies. Even simpler methods such as the rule-based classifier, function, and decision trees achieved similar accuracy results, joining the prior algorithms with values higher than 90%. Overall the support vector machine could be considered to have the best performance with the highest accuracy and sensitivity at 91.5% and 6.4% respectively, as well as achieving perfect specificity and a zero false positive rate. The strength of the support vector machine algorithm may be attributed the fact that it regulates the number of

Algorithm	Accuracy	Sensitivity	Specificity	Precision	False positive rate	F-measure	F_2
GP1	0.606	0.493	0.618	0.114	0.382	0.185	0.296
GP2	0.910	0.025	0.998	0.548	0.002	0.049	0.031
GA1	0.910	0.008	1.000	0.636	0.000	0.015	0.010
GA2	0.911	0.016	1.000	0.778	0.000	0.031	0.019
ACO	0.892	0.032	0.978	0.125	0.022	0.051	0.038
NN1	0.771	0.271	0.821	0.131	0.179	0.176	0.223
NN2	0.774	0.193	0.832	0.103	0.168	0.134	0.164
SVM	0.915	0.064	1.000	0.951	0.000	0.120	0.079
FL	0.909	0.019	0.998	0.459	0.002	0.036	0.023
DT1	0.903	0.073	0.986	0.338	0.014	0.120	0.087
DT2	0.909	0.017	0.998	0.405	0.002	0.032	0.021
Fn	0.910	0.004	1.000	0.667	0.000	0.009	0.006
Lazy	0.652	0.504	0.667	0.131	0.333	0.208	0.321
Rule	0.910	0.032	0.998	0.592	0.002	0.061	0.040

Table 4: Popular binary classification performance metrics for results on the UCSD dataset

Algorithm	Accuracy	Sensitivity	Specificity	Precision	False positive rate	F-measure	F_2
GP1	0.014	0.047	0.015	0.010	0.015	0.016	0.026
GP2	0.001	0.022	0.003	0.247	0.003	0.030	0.021
GA1	0.001	0.012	0.001	0.415	0.001	0.009	0.006
GA2	0.002	0.013	0.001	0.200	0.001	0.016	0.011
ACO	0.023	0.033	0.027	0.152	0.027	0.029	0.027
NN1	0.023	0.038	0.027	0.016	0.027	0.018	0.025
NN2	0.057	0.094	0.071	0.023	0.071	0.030	0.053
SVM	0.002	0.022	0.001	0.075	0.001	0.039	0.027
FL	0.002	0.012	0.002	0.273	0.002	0.020	0.013
DT1	0.004	0.027	0.004	0.088	0.004	0.040	0.031
DT2	0.002	0.024	0.003	0.162	0.003	0.041	0.028
Fn	0.001	0.005	0.001	0.400	0.001	0.000	0.000
Lazy	0.018	0.040	0.020	0.009	0.020	0.015	0.023
Rule	0.003	0.021	0.002	0.292	0.002	0.037	0.025

Table 5: Standard deviation of performance metrics over the UCSD dataset

support vectors, allowing finer control of the experiment (Schölkopf et al., 2000).

Both of the genetic algorithm variants, GA1 and GA2, allowed for the possibility of failing a sample by not being able to confidently classify it into either class. This was caused by the fitness function they applied to each generated rule resulting in a tie between both classes (Guan and Zhu, 2005). Therefore, in our results unclassified samples for these two algorithms were removed from consideration in the results reported above. Despite this indecision both algorithms achieved reasonable success including perfect specificity on the credit card dataset.

Though some algorithms had significantly lower accuracy results than the trend set by the majority these methods seem to make up for it with a much higher sensitivity. It may be the case that some of the techniques learned that the ratio of classes was unbalanced enough that skewing their classification to the negative resulted in better overall performance. However, other algorithms such as GP1, both neural networks, and the lazy evaluator all had much higher sensitivities, indicating a deeper understanding of the problem. The reason for their differing behaviour may be due to them being built incrementally so that higher sensitivity results are encouraged early on in their learning. GP1s fitness function, for example, deliberately used both sensitivity and specificity in its evaluation of each generation (Bojarczuk et al., 2004).

Figure 1 shows some interesting behaviour between different types of algorithms. Unsurprisingly, both genetic algorithms and neural networks had relatively consistent error rates, demonstrating their similarities. However the results were vastly different between both genetic programming techniques. GP1 specifically used a simpler approach with its constrained-syntax model, and it may be the case that this restrictiveness is less successful than the unconstrained rule generation used by GP2 (Bojarczuk et al., 2004), (De Falco et al., 2002).

We can also observe from Figure 1 that our earlier analysis of detection algorithm performance was correct. The GP1, neural network, and lazy evaluator algorithms all demonstrated a much higher average error rate than the very consistent standard set by the rest of the methods. The graph also indicates that the ant colony optimisation and neural networks both had higher variance between fold instances than the remaining detection algorithms.

From the evidence in Table 5 we can see that the ant colony optimisation and both neural networks all tended to a higher standard deviation across metrics than other methods. Specifically NN2 had the highest out of all methods, achieving 0.094 on sensitivity and 0.071 on specificity. These results show a correlation with the higher variance in error rates found between these three methods earlier in Figure 1, and supports our claim that the iterative structure of these algorithms leads to the chance of small variance during the initial stages compounding into larger differences by completion. The receptiveness of the ant colony optimisation and neural network methods to small differences between each fold makes them suitable algorithms for financial fraud detection.

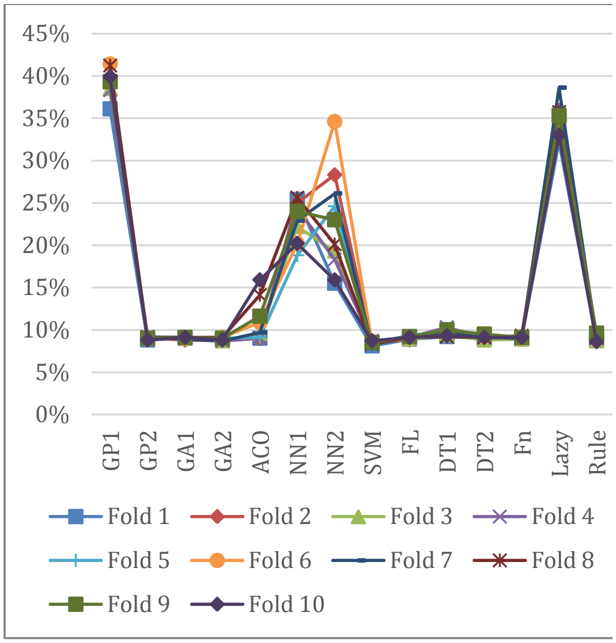

Figure 1: Percentage of classification error for each method per dataset fold

4.2 Feature Selection

The similarity values given in Table 6 are partly derived from the fact that there was a core subset of four or five common features chosen by the majority of algorithms. The remaining attributes were mostly disparate, indicating that they were more difficult to differentiate between in relevance to the problem. The fact that these algorithms had very similar results despite this disparity suggests that the vast majority of these attributes did not contribute to the characteristics of the problem.

Table 6 also shows that there was a considerable overlap between features selected for several algorithms, which is understandable. However, FS6 had noticeably fewer shared attributes, no more than two in common with any other algorithm, which was reflected in its results. GP2, GA1, and GA2 all achieved 0% sensitivity when performed on the features selected by FS6. This could indicate that FS6 has selected inferior attributes than the rest of the methods. However, it also had relatively high performance on NN1 with the second highest accuracy and reasonable performance with the remaining detection algorithms which may suggest that the attributes it chose allow for more variance in their execution.

Methods	FS1	FS2	FS3	FS4	FS5	FS6	FS7
FS1	■	28.57	42.86	50.00	28.57	14.29	57.14
FS2	28.57	■	28.57	50.00	28.57	7.14	50.00
FS3	42.86	28.57	■	35.71	21.43	7.14	71.43
FS4	50.00	50.00	35.71	■	35.71	14.29	50.00
FS5	28.57	28.57	21.43	35.71	■	0.00	35.71
FS6	14.29	7.14	7.14	14.29	0.00	■	7.14
FS7	57.14	50.00	71.43	50.00	35.71	7.14	■

Table 6: Percentage of attribute similarity between the results of each feature selection algorithm

There was some unexpected differences between several of the detection algorithms when executed on the different feature sets. Most of the detection algorithms had comparable performances on each set of features, with accuracy values only differing by less than 1.3%. However, the algorithms that demonstrated the largest sensitivities in section 4.1 had much higher variances in results between feature sets, with accuracies varying from 7.0% to as high as 19.2%. This may indicate that the underlying relationships between the features can be uncovered better with detection algorithms that are more incremental in nature.

4.3 Performance Metrics

As discussed in prior sections fraud detection has a significant imbalance in ratio of positive to negative instances, but it also has a high, inverse ratio of costs between the two. If this disparity is large enough the higher sensitivity and lower accuracy exhibited by some of the detection methods may indicate a better approach than other metrics which look nominally superior. Its ability to handle these misclassification results could indicate that cost minimisation is a valid metric for the problem. In addition to error rates Figure 1 demonstrates the consistency between folds on the dataset. Most algorithms had comparable results but the ant colony optimisation and neural networks both had significant disparity. The potential cause of this is the iterative method by which these algorithms learn allowing for greater initial deviation which leads to a more varied overall result. Because each fold is a subset of the same dataset large variances in the results between folds can indicate a higher sensitivity to minor changes in the data, which is a desirable trait in a complicated problem like financial fraud detection. Table 5 indicates that precision showed the greatest standard deviation out of any metric, reaching as high as 0.415 for GA1 and 0.4 for Fn. This is most likely another consequence of the imbalanced nature of the dataset, which has resulted in a large difference in the ratio of true positives to true negatives and therefore a larger variance for the metrics that analyses them. As this imbalance is typical for the financial fraud detection problem precision is likely a superior metric for assessing algorithm performance.

5 Conclusion

This paper investigated several important fraud detection experimental issues including problem representation, detection algorithms, feature selection, and performance metrics. Our purpose is to provide a reference for future researchers and practitioners to utilise when undertaking their own experiments. We also studied some of these experimental issues more directly by conducting several experiments with a focus detection algorithms, feature selection, and performance metrics for credit card fraud. We observed that many detection algorithms demonstrated performance in one or two areas and that it’s necessary for the experimenter to determine which of these should be assigned a higher priority. For example, if misclassification costs are high, techniques with a higher sensitivity such as GP1, neural networks, or the lazy evaluator may be suitable choices. If receptiveness to minor variances in the dataset is desired then the ant colony optimisation or neural networks could be appropriate. The addition of feature selection experimentation provided an additional viewpoint for our simulations, and we observed that a subset of common attributes were chosen by the majority of selection algorithms indicating that they were significantly representative of the problem characteristics. When studying performance metrics we identified that precision detected deviation in the dataset better than other metrics, indicating that it is more sensitive to the underlying relationships and may be a valid metric for studying financial fraud detection. Of course, we acknowledge that the effectiveness of these experiments is determinant on the individual experimental details, and these may change dramatically for different fraud problems or requirements. There are several areas that future researchers could focus on, most obviously exploring each of the experimental issues covered here in the context of a single fraud type or detection algorithm. Additionally they could focus on further comparisons between different metrics and methods using a real-world dataset or expand on the feature selection aspect of fraud detection with a specific focus on the usefulness of individual attributes.

References

Bhattacharyya S, Jha S, Tharakunnel K, and Westland JC (2011) Data mining for credit card fraud: A comparative study. Decision Support Systems 50, 602-13.

Bhattacharya M, "Reduced Computation for Evolutionary Optimization in Noisy Environment", in Proceedings of ACM Genetic and Evolutionary Computation Conference 2008 (GECCO 2008),

Bhattacharya M, Islam R and Abawajy J (2016) Evolutionary Optimization: A Big Data Perspective, The Journal of Network and Computer Applications, Elsevier, ISSN: 1084-8045, Vol. 59, pp. 416-426, 2016.

Atlanta, USA, ACM Press, ISBN: 978-1-60558-131-6, pp. 2117-2122.

Bojarczuk CC, Lopes HS, Freitas AA, and Michalkiewicz EL (2004) A constrained-syntax genetic programming system for discovering classification rules: application to medical data sets. Artificial Intelligence in Medicine 30, 27-48.

Dal Pozzolo A, Caelen O, Le Borgne Y-A, Waterschoot S, and Bontempi G (2014) Learned lessons in credit card fraud detection from a practitioner perspective. Expert systems with applications 41, 4915-28.

De Falco I, Della Cioppa A, and Tarantino E (2002) Discovering interesting classification rules with genetic programming. Applied Soft Computing 1, 257-69.

Duman E and Ozcelik MH (2011) Detecting credit card fraud by genetic algorithm and scatter search. Expert Systems with Applications 38, 13057-63.

Fawcett T (2006) An introduction to ROC analysis. Pattern recognition letters 27, 861-74.

Guan S-U and Zhu F (2005) An incremental approach to genetic-algorithms-based classification. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 35, 227-39.

Hall M and Holmes G (2003), Benchmarking attribute selection techniques for discrete class data mining, Knowledge and Data Engineering, IEEE Transactions on, 2003, 15, (6), pp. 1437-1447.

Halvaiee NS and Akbari MK (2014) A novel model for credit card fraud detection using Artificial Immune Systems. Applied Soft Computing 24, 40-9.

Han J, Kamber M, and Pei J (2011) In Data mining: concepts and techniques. Vol. pp. Elsevier,

Kantardzic M (2011) In Data mining: concepts, models, methods, and algorithms. Vol. pp. John Wiley & Sons,

Ngai E, Hu Y, Wong Y, Chen Y, and Sun X (2011) The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. Decision Support Systems 50, 559-69.

Olszewski D (2014) Fraud detection using self-organizing map visualizing the user profiles. Knowledge-Based Systems 70, 324-34.

Panigrahi S, Kundu A, Sural S, and Majumdar AK (2009) Credit card fraud detection: A fusion approach using Dempster–Shafer theory and Bayesian learning. Information Fusion 10, 354-63.

Quah JT and Sriganesh M (2008) Real-time credit card fraud detection using computational intelligence. Expert Systems with Applications 35, 1721-32.

Ravisankar P, Ravi V, Rao GR, and Bose I (2011) Detection of financial statement fraud and feature selection using data mining techniques. Decision Support Systems 50, 491-500.

Sahin Y, Bulkan S, and Duman E (2013) A cost-sensitive decision tree approach for fraud detection. Expert Systems with Applications 40, 5916-23.

Sánchez D, Vila M, Cerda L, and Serrano J-M (2009) Association rules applied to credit card fraud detection. Expert Systems with Applications 36, 3630-40.

Schölkopf B, Smola AJ, Williamson RC, and Bartlett PL (2000) New support vector algorithms. Neural computation 12, 1207-45.

West J, Bhattacharya M and Islam R (2014) Intelligent Financial Fraud Detection Practices: An Investigation", in Proceedings of the 10th International Conference on Security and Privacy in

Communication Networks (SecureComm 2014), Vol. 153, 2015, LNICS, Springer, ISBN: 978-3-319-23801-2 (Print) 978-3-319-23802-9 (Online), pp. 186-203.

West J and Bhattacharya M (2015a) Some Experimental Issues in Financial Fraud Detection: An Investigation, Proceedings of The 5th International Symposium on Cloud and Service Computing (SC2 2015), IEEE CS Press.

West J and Bhattacharya M (2015b) Mining Financial Statement Fraud: An Analysis of Some Experimental Issues, Proceedings of The 10th IEEE Conference on Industrial Electronics and Applications (ICIEA 2015), IEEE Press, ISBN: 978-1-4673-7317-3, pp. 461-466.

West J and Bhattacharya M (2016a) Intelligent financial fraud detection: a comprehensive review. Computers & Security, Elsevier, ISSN: 0167-4048, Vol. 57, pp. 47-66, 2016.

West J and Bhattacharya M (2016b) An Investigation on Experimental Issues in Financial Fraud Mining, Proceedings of The 11th IEEE Conference on Industrial Electronics and Applications (ICIEA 2016), IEEE Press.

Wong N, Ray P, Stephens G, and Lewis L (2012) Artificial immune systems for the detection of credit card fraud: an architecture, prototype and preliminary results. Information Systems Journal 22, 53-76.

Yang Q and Wu X (2006) 10 challenging problems in data mining research. International Journal of Information Technology & Decision Making 5, 597-604.

Zhou W and Kapoor G (2011) Detecting evolutionary financial statement fraud. Decision Support Systems 50, 570-5.