Minority Report in Fraud Detection: Classification of Skewed Data

Clifton Phua, Damminda Alahakoon, and Vincent Lee

School of Business Systems, Faculty of Information Technology
Monash University, Clayton campus
Wellington Road, Clayton, Victoria 3800, Australia

{clifton.phua,damminda.alahakoon,cheng.lee}@infotech.monash.edu

ABSTRACT

This paper proposes an innovative fraud detection method, built upon existing fraud detection research and *Minority Report*, to deal with the data mining problem of skewed data distributions. This method uses backpropagation (BP), together with naive Bayesian (NB) and C4.5 algorithms, on data partitions derived from minority oversampling with replacement. Its originality lies in the use of a single meta-classifier (stacking) to choose the best base classifiers, and then combine these base classifiers' predictions (bagging) to improve cost savings (stacking-bagging). Results from a publicly available automobile insurance fraud detection data set demonstrate that stacking-bagging performs slightly better than the best performing bagged algorithm, C4.5, and its best classifier, C4.5 (2), in terms of cost savings. Stacking-bagging also outperforms the common technique used in industry (BP without both sampling and partitioning). Subsequently, this paper compares the new fraud detection method (meta-learning approach) against C4.5 trained using undersampling, oversampling, and SMOTEing without partitioning (sampling approach). Results show that, given a fixed decision threshold and cost matrix, the partitioning and multiple algorithms approach achieves marginally higher cost savings than varying the entire training data set with different class distributions. The most interesting find is confirming that the combination of classifiers to produce the best cost savings has its contributions from all three algorithms.

Keywords

Fraud detection, multiple classifier systems, meta-learning

1. INTRODUCTION

Fraud, or criminal deception, will always be a costly problem for many profit organisations. Data mining can minimise some of these losses by making use of the massive collections of customer data, particularly in insurance, credit card, and telecommunications industries.

However, fraud detection data being highly skewed or imbalanced is the norm. Usually there are many more legitimate than fraudulent examples. This means that by predicting all instances to be legal, a very high success rate is achieved without detecting any fraud.

There can be two typical ways to proceed when faced with this problem. The first approach is to apply different algorithms (meta-learning). Each algorithm has its unique strengths, so that it may perform better on particular data instances than the rest [41]. The second approach is to manipulate the class distribution (sampling). The minority class training examples can be increased in proportion to the majority class in order to raise the chances of correct predictions by the algorithm(s).

Most of the published work on improving the performance of standard classifiers on skewed data usually involves using the same algorithm(s). For example, the work on cost sensitive learning [7; 33; 14] aimed at reducing total cost, and sampling approaches [14; 24; 9] to favour the minority class are usually demonstrated with decision tree algorithms and/or naive Bayes. This paper introduces the new fraud detection method to predict criminal patterns from skewed data:

- The innovative use of naive Bayesian (NB), C4.5, and backpropagation (BP) classifiers to process the same partitioned numerical data has the potential of getting better cost savings.
- The selection of the best classifiers of different algorithms using stacking and the merger of their predictions using bagging is likely to produce better cost savings than either bagging multiple classifiers from same algorithm, bagging each algorithm's bagged result , stacking all classifiers, or choosing the best classifier approaches.

One related problem caused by skewed data includes measuring the performance of the classifiers. Success cannot be defined in terms of predictive accuracy because the minority class in the skewed data usually has a significantly higher cost.

Recent work on skewed data sets was evaluated using better performance metrics such as Area Under Curve (AUC) [9, 10], cost curves [15], and Receiver Operating Characteristic (ROC) analysis [28]. But this paper chooses a simplified cost model to detect insurance fraud, adapted from credit card fraud [8], to concentrate on the viability of the fraud detection method.

Section 2 contains existing fraud detection methods; the new fraud detection method; the reasons for the choice of the three classification algorithms; and introduces the hybrid ensemble mechanism.

Section 3 briefly describes the experimental data set; defines the cost model; provides the rationale for creating derived attributes and explains the data preparation; details the partitioning and oversampling strategy used; and describes the experimental plan.

Section 4 provides the results on highest cost savings from the experiments. Section 5 discusses the main lessons learnt from the

experiments. Section 6 highlights the limitations. Section 7 considers the possible future work and Section 8 concludes the paper.

2. FRAUD DETECTION

2.1 Existing Fraud Detection Methods

This subsection concentrates on the analysis of some reliable data mining methods applied specifically to the data-rich areas of insurance, credit card, and telecommunications fraud detection, in order to integrate some of them. A brief description of each method and its applications is given.

2.1.1 Insurance Fraud

[29] recommends the use of dynamic real-time Bayesian Belief Networks (BBNs), named Mass Detection Tool (MDT), for the early detection of potentially fraudulent claims, that is then used by a rule generator named Suspicion Building Tool (SBT). The weights of the BBN are refined by the rule generator's outcomes and claim handlers have to keep pace with evolving frauds. This approach evolved from ethnology studies of large insurance companies and loss adjustors who argued against the manual detection of fraud by claim handlers.

The hot spot methodology [37] applies a three step process: the k-means algorithm for cluster detection, the C4.5 algorithm for decision tree rule induction, and domain knowledge, statistical summaries and visualisation tools for rule evaluation. It has been applied to detect health care fraud by doctors and the public for the Australian Health Insurance Commission. [38] has expanded the hot spot architecture to use genetic algorithms to generate rules and to allow the domain user, such as a fraud specialist, to explore the rules and to allow them to evolve according to how interesting the discovery is. [4] presented a similar methodology utilising the Self Organising Map (SOM) for cluster detection before BP neural networks in automobile injury claims fraud.

The use of supervised learning with BP neural networks, followed by unsupervised learning using SOM to analyse the classification results, is recommended by [22]. Results from clustering show that, out of the four output classification categories used to rate medical practice profiles, only two of the well defined categories are important. Like the hotspot methodology, this innovative approach was applied on instances of the Australian Health Insurance Commission health practitioners' profiles.

2.1.2 Credit Card Fraud

The Bayesian Belief Network (BBN) and Artificial Neural Network (ANN) comparison study [27] uses the STAGE algorithm for BBNs and BP algorithm for ANNs in fraud detection. Comparative results show that BBNs were more accurate and much faster to train, but BBNs are slower when applied to new instances. Real world credit card data was used but the number of instances is unknown.

The distributed data mining model [8] is a scalable, supervised black box approach that uses a realistic cost model to evaluate C4.5, CART, Ripper and NB classification models. The results demonstrated that partitioning a large data set into smaller subsets to generate classifiers using different algorithms, experimenting with fraud:legal distributions within training data and using stacking to combine multiple models significantly improves cost savings. This method was applied to one million credit card transactions from two major US banks, Chase Bank and First Union Bank.

FairIsaac, formerly known as HNC, produces software for detecting credit card fraud. It favours a three-layer BP neural network for processing transactional, cardholder, and merchant data to detect fraudulent activity [36].

2.1.3 Telecommunications Fraud

The Advanced Security for Personal Communications Technologies (ASPECT) research group [36] focuses on neural networks, particularly unsupervised ones, to train legal current user profiles that store recent user information and user profile histories that store long term information to define normal patterns of use. Once trained, fraud is highly probable when there is a difference between a mobile phone user's current profile and the profile history.

Cahill *et al* [5] builds upon the adaptive fraud detection framework [20] by using an event-driven approach of assigning fraud scores to detect fraud as it happens, and weighting recent mobile phone calls more heavily than earlier ones. The new framework [5] can also detect types of fraud using rules, in addition to detecting fraud in each individual account, from large databases. This framework has been applied to both wireless and wire line fraud detection systems with over two million customers.

The adaptive fraud detection framework presents rule-learning fraud detectors based on account-specific thresholds that are automatically generated for profiling the fraud in an individual account. The system, based on the framework, has been applied by combining the most relevant rules, to uncover fraudulent usage that is added to the legitimate use of a mobile phone account [19; 20].

2.2 The New Fraud Detection Method

This subsection proposes a different but non-trivial method of detecting crime based partially on the science fiction novel, *Minority Report* [12]. The idea is to simulate the book's Precrime method of precogs and integration mechanisms with existing data mining methods and techniques. An overview of how the new method can be used to predict fraud for each instance is provided.

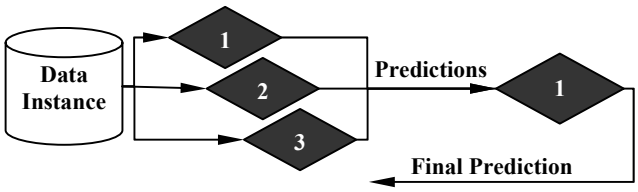

Figure 1: Predictions on a single data instance using precogs

2.2.1 Precogs

Precogs, or precognitive elements, are entities that have the knowledge to predict that something will happen. Figure 1 above uses three precogs, labelled 1, 2, and 3, to foresee and prevent crime by stopping potentially guilty criminals [12]. Unlike the human "mutant" precogs [12], each precog contains multiple black-box classification models, or classifiers, trained with one

data mining technique in order to extrapolate the future. The three precogs proposed here are different from each other in that they are trained by different data mining algorithms. For example, the first, second, and third precog are trained using the statistical paradigm (NB), computer metaphor (C4.5) and brain metaphor (BP) respectively. They require numerical inputs of past examples to output corresponding class predictions for new instances.

2.2.2 Integration Mechanisms

Figure 1 shows that as each precog outputs its many predictions for each instance, all the predictions are fed back into one of the precogs, to derive a final prediction for each instance.

2.3 Fraud Detection Algorithms

This subsection presents a summary of each algorithm's main advantages and disadvantages. The qualitative and quantitative justification for using the three different techniques together on the same fraud data is also highlighted. This subsection also advocates another cross validation approach to preparing data for training. The use of bagging for combining predictions from one algorithm, stacking for combining predictions from all algorithms, and its hybrid is evaluated.

2.3.1 Classifiers

- Although the naive Bayesian (NB) algorithm is simple, it is very effective in many real world data sets because it can give better predictive accuracy than well known methods like C4.5 and BP [13; 18] and is extremely efficient in that it learns in a linear fashion using ensemble mechanisms, such as bagging and boosting, to combine classifier predictions [17]. However, when attributes are redundant and not normally distributed, the predictive accuracy is reduced [39].
- C4.5 can help not only to make accurate predictions from the data but also to explain the patterns in it. It deals with the problems of the numeric attributes, missing values, pruning, estimating error rates, complexity of decision tree induction, and generating rules from trees [39]. In terms of predictive accuracy, C4.5 performs slightly better than CART and ID3 [31]. C4.5's successor, C5.0, shows marginal improvements to decision tree induction but not enough to justify its use. The learning and classification steps of C4.5 are generally fast [21]. However, scalability and efficiency problems, such as the substantial decrease in performance and poor use of available system resources, can occur when C4.5 is applied to large data sets.
- Backpropagation (BP) neural networks can process a very large number of instances; have a high tolerance to noisy data; and has the ability to classify patterns which they have not been trained [21]. They are an appropriate choice if the results of the model are more important than understanding how it works [1]. However, the BP algorithm requires long training times and extensive testing and retraining of parameters, such as the number of hidden neurons, learning rate and momentum, to determine the best performance [2].

2.3.2 Justification of Algorithms

Table 1 below summarises the preceding section to qualitatively show that each algorithm is intrinsically different from one another. Effectiveness highlights the overall predictive accuracy and performance of each algorithm. Scalability refers to the capability to construct a model effectively given large data sets. Speed refers to efficiency in model construction.

Table 1: Qualitative comparison of algorithms

Algorithm	Effectiveness	Scalability	Speed
NB	Good	Excellent	Excellent
C4.5	Excellent	Poor	Good
BP	Good	Excellent	Poor

The strongest quantitative arguments to justify for three different algorithms and in the form of NB, C4.5 and BP come from [16] and [25] in recent meta-learning literature. [16] showed that three base-level classifiers perform comparably, if not better than seven base-level classifiers (each classifier was computed with a different algorithm). To discover the diversity between classification algorithms, [25] used ten different ones and proved that a simple version of NB exhibits the most different behaviour compared to C5.0 and a similar form of BP.

To discover diversity between data sets using ranks, [25] clusters eighty different ones into four groups and applied the ten algorithms on each data set. It was discovered that in the first cluster of eighteen data sets, a similar form of BP was one of the better performers while NB performed worst; in the second cluster of twenty four data sets, C5.0 offered the best performance while NB and a similar form of BP performed the worst; in the third cluster of ten data sets, C5.0 still offered the best performance while two similar forms of BP performed worst; and in the fourth cluster of twenty eight data sets, most of the algorithms did not perform significantly different from each other.

Therefore, by using the three algorithms together on the same skewed data, within the context of classification analysis, their strengths can be combined and their weaknesses reduced. Also, these three algorithms promise the best predictive capability in fraud detection compared to other classification algorithms. *K-nearest* neighbour, case-based reasoning, genetic algorithms, and rough sets algorithms either have scalability problems or are still in their prototype phase [21].

2.3.3 Combining Output

This study provides a slight variation of cross validation. Instead of using ten data partitions, an odd-numbered eleven data partitions are used so that there will always be a majority class when the partitions contribute their class vote.

- Bagging [3] combines the classifiers trained by the same algorithm using unweighted majority voting on each example or instance. Voting denotes the contribution of a single vote, or its own prediction, from a classifier. The final prediction is then decided by the majority of the votes. Generally, bagging performs significantly better than the single model for C4.5 and BP algorithms. It is never substantially worse because it

neutralises the instability of the classifiers by increasing the success rate [39].

- Stacking [40] combines multiple classifiers generated by different algorithms with a meta-classifier. To classify an instance, the base classifiers from the three algorithms present their predictions to the meta-classifier which then makes the final prediction.
- Stacking-bagging is a hybrid technique proposed by this paper. The recommendation here is to train the simplest learning algorithm first, followed by the complex ones. In this way, NB base classifiers are computed, followed by the C4.5 and then the BP base classifiers. The NB predictions can be quickly obtained and analysed while the other predictions, which take longer training and scoring times, are being processed. As most of the classification work has been done by the base classifiers, the NB algorithm, which is simple and fast, is used as the meta-classifier [8]. In order to select the most reliable base classifiers, stacking-bagging uses stacking to learn the relationship between classifier predictions and the correct class. For a data instance, these chosen base classifiers' predictions then contribute their individual votes and the class with the most votes is the final prediction.

3. EXPERIMENTS

3.1 Data Understanding

The only available fraud detection data set in automobile insurance is provided by Angoss KnowledgeSeeker software. Originally named "carclaims.txt", it can be found in the accompanying compact disc from [34]. Experiments described in this paper split the main data set into a training data set and a scoring data set. The class labels of the training data are known, and the training data is historical compared to the scoring data. The class labels of the score data set are removed, and the score data set is then processed by the classifiers for actual predictions.

This data set contains 11338 examples from January 1994 to December 1995 (training data), and 4083 instances from January 1996 to December 1996 (score data). It has a 6% fraudulent and 94% legitimate distribution, with an average of 430 claims per month. The original data set has 6 numerical attributes and 25 categorical attributes, including the binary class label (fraud or legal).

The data quality is good but there are some impediments. The original data set consists of the attribute *PolicyType* (discarded) which is an amalgamation of existing attributes *VehicleCategory* and *BasePolicy*. There are invalid values of 0 in each of the attributes *MonthClaimed* and *DayofWeekClaimed* for one example (deleted). Some attributes with two categories, like *WitnessPresent*, *AgentType*, and *PoliceReportFiled*, have highly skewed values where the minority examples account for less than 3% of the total examples (unchanged). The attribute *Make* has a total of 19 possible attribute values of which claims from Pontiac, Toyota, Honda, Mazda, and Chevrolet account for almost 90% of the total examples (unchanged). There are three spelling mistakes in Make (corrected): *Accura* (Acura), *Mecedes* (Mercedes), *Nisson* (Nissan), and *Porche* (Porsche).

3.2 Cost Model

There is a need to measure the benefit of detecting fraud and this particular cost model has two assumptions. First, all alerts must be investigated. Second, the average cost per claim must be higher than the average cost per investigation. Taking the year 1996 into account, the average cost per claim for the score data set is approximated at USD$2,640 [23] and average cost per investigation is estimated at USD$203 for ten manpower hours [30].

Table 2 below illustrates that hits and false alarms require investigation costs; and misses and normals pay out the usual claim cost. False alarms are the most expensive because they incur both investigation and claim costs.

Table 2: Cost model for insurance fraud detection

Outcome	Cost
Hits	Number of Hits * Average Cost Per Investigation
False Alarms	Number of False Alarms * (Average Cost Per Investigation + Average Cost Per Claim)
Misses	Number of Misses * Average Cost Per Claim
Normals	Number of Normal Claims * Average Cost Per Claim

Table 3 below shows that there are two extremes of this model: at one end, data mining is not used at all (no action), so all claims are regarded as normals; at the other end, data mining achieves the perfect predictions (best case scenario), so all claims are predicted as hits and normals. Therefore the evaluation metrics for the predictive models on the score data set to find the optimum cost savings are:

Model Cost Savings = No Action – [Misses Cost + False Alarms Cost + Normals Cost + Hits Cost]

Percentage Saved = (Model Cost Savings / Best Case Scenario Cost Savings * 100)

Table 3: Cost matrix for fraud detection

Prediction	Fraud	Legal
Alert	Hits	False Alarms
No alert	Misses	Normals

3.3 Data Preparation

3.3.1 Construct the Data

Three derived attributes, *weeks_past*, is_*holidayweek_claim*, and *age_price_wsum*, are created to increase predictive accuracy for the algorithms. The new attribute, *weeks_past*, represents the time difference between the accident occurrence and its claim application. The position of the week in the year that the claim was made is calculated from attributes *month_claimed*, *week_of_month_claimed*, and *year*. Then the position of the week in the year when the accident is reported to have happened is

computed from attributes *month*, *week_of_month*, and *year*. The latter is subtracted from the former to obtain the derived attribute *weeks_past*. This derived attribute is then categorised into eight discrete values.

The derived attribute *is_holidayweek_claim* indicates whether the claim was made in a festive week [1]. There is a speculation that average offenders are more likely to strike during those weeks because they will want to spend more money and probably believe their chances of getting caught is lower. A major assumption about this data set being from the US has to be made. Therefore, in the years 1994 and 1995, the attribute *is_holidayweek_claim* is set to 1 if the claim is made during a week containing at least one US public holiday. This computed attribute is binary-valued.

The attribute *age_price_wsum* is the weighted sum of two related attributes, *age_of_vehicle* and *vehicle_price*. The assumption is that if the vehicle gets older and its current value is still expensive, then the possibility of the claim being fraudulent becomes higher. This derived attribute has seven discrete values.

The input data must be the same for all three algorithms so that predictions can be compared and combined. The NB and C4.5 algorithms can train with both numeric and non-numeric data. However, the BP algorithm must always train with numeric data. Due to this incompatibility, all training examples are scaled to numbers between 0 and 1, or transformed into one-out-of-N and binary encodings which are between 0 and 1. One-of-N coding is used to represent a unique set of inputs and is done by having a length equal to the number of discrete values for the attribute. For example, there are the values 1994, 1995 and 1996 for attribute Year which are represented by 1 0 0, 0 1 0, and 0 0 1 respectively. This coding is simple to understand and easy to use. However, it is not suitable for attributes with a large number of values. Binary coding overcomes the limitation of one-of-N coding but has increased complexity by representing each discrete value with a string of binary digits [2]. There are twelve values for attribute month can be represented with a binary code vector of length 4 (16 possible values). For example, the attribute values, January and December, are converted to 0 0 0 1 and 1 1 0 0 respectively. There is a significant consequence of this new data requirement: all attributes must be treated as non-numeric for the NB and C4.5 algorithms, and numeric for the BP algorithm.

For this data set, fourteen attributes are scaled in the range 0 to 1. Nineteen attributes with no logical ordering are represented by either one-of-N or binary coding.

3.3.2 Partition the Data

According to [8], the desired distribution of the data partitions belonging to a particular fraud detection data set must be determined empirically. In a related study, it is recommended by [6; 31] that data partitions should neither be too large for the time complexity of the learning algorithms nor too small to produce poor classifiers.

Given this information, the approach adopted in this thesis is to randomly select different legal examples from the years 1994 and 1995 (10840 legal examples) into eleven sets of *y* legal examples (923). The data partitions are formed by merging all the available *x* fraud examples (615) with a different set of *y* to form eleven *x:y* partitions (615:923) with a fraud:legal distribution of 40:60. Other possible distributions are 50:50 (923:923) and 30:70 (396:923). Therefore, there are eleven data partitions with 1538 examples, another set of eleven data partitions with 1846 examples, and the last eleven data partitions with 1319 examples. [8] contains a more detailed discussion of this partitioning approach. Skewed data is transformed into partitions with more of the rarer examples and fewer of the common examples using the procedure known as minority oversampling with replacement/replication [24].

In rotation, each data partition of a certain distribution is used for training, testing and evaluation once. A training data partition is used to come up with a classifier, a test data partition to optimise the classifier's parameters and an evaluation data partition to compare the classifier with others. All the data partitions used by a single classifier are independent of each other. All classifiers must be trained before they are scored.

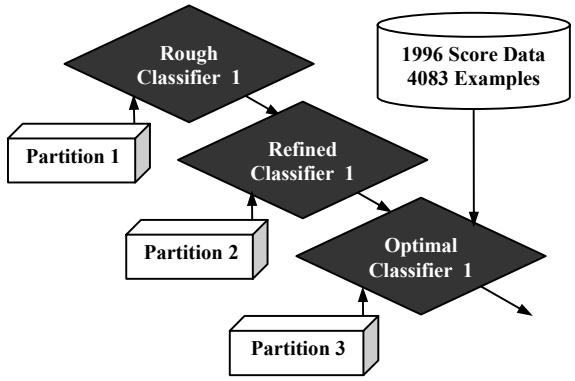

Figure 2: Building and applying classifier 1 using data partitions*
***Source:** adapted from [1]

Figure 2 above demonstrates that the algorithm is first trained on partition 1 to generate classifier 1, tested on partition 2 to refine the classifier and evaluated on partition 3 to assess the expected accuracy of the classifier (Test A). The next session is training on partition 2 to generate classifier 2, tested on partition 3, and evaluated on partition 4 (Test B). This continues until there are eleven cross validated training sessions with eleven classifiers. The classifiers are then applied to the score data set. Their corresponding success rate estimates and class predictions are recorded for further analysis (see Table 4).

[32] points out the criticism that duplicating the minority examples does not add any new information into the data and it pales in comparison to adjusting the output threshold correctly. To substantiate the cause for sampling, [28] produced some evidence showing that sampling does produce the same effect as moving the decision threshold or adjusting the cost matrix. Also, if the highly skewed data (for example, 6% minority class) is not sampled to balance the class distribution, some current data mining software (for example, NB and C4.5 software) with non-numeric output (in the form of either "fraud" or "legal" with a 0.5 threshold) will fail to detect any fraud. Therefore, no cost savings will be achieved.

3.4 Modelling

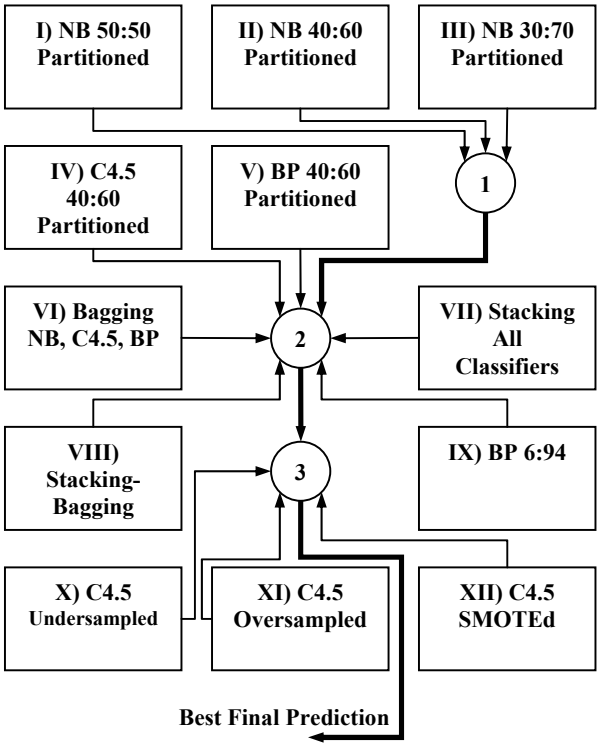

Figure 3: Experiments overview

Figure 3 above lists the nine experiments which were based on the meta-learning approach, and another three experiments which utilised the sampling approach. Each rectangle represents an experiment, describes the learning algorithm used and the fraud:legal data distribution. Each circle depicts a comparison of cost savings between experiments. Each bold arrow indicates the best experiment from the comparisons. Decision threshold (except for Experiments V and IX) and cost model for these experiments will remain unchanged. Experiments V and IX will produce BP predictions which are between 0 and 1, and these numerical predictions need to be converted into categorical ones using the decision threshold value which maximises the cost savings.

in the next column, Table 4 lists the eleven tests, labelled A to K, which were repeated for each of Experiments I to V. In other words, there are fifty five tests in total for Experiments I to V. Each test consisted of training, testing, evaluation, and scoring (see Section 3.3.2). The score set was the same for all classifiers but the data partitions labelled 1 to 11 were rotated. The overall success rate denotes the ability of an ensemble of classifiers to provide correct predictions. The bagged overall success rates *X* and *Z* were compared to the averaged overall success rates *W* and *Y*. The predictions for each of the first five experiments were obtained by bagging the eleven predictions on the score set, represented by *Bagged Z* (see Table 4).

Table 4: The tests plan for the Experiments I to V

Training and Scoring	Test A	Tests B to K	Overall Success Rate
Training Set	Partition 1	2 to 11	
Testing Set	Partition 2	3 to 12	
Evaluation Set	Partition 3	4 to 2	
Evaluating	*Success Rate A*	*B to K*	*Average W*
Bagging	Predictions A	B to K	*Bagged X*
Producing	Classifier 1	2 to 11	
Scoring Set	*Success Rate A*	*B to K*	*Average Y*
Bagging Score Set	Score Predictions A	B to K	*Bagged Z*

Experiments I, II and III were designed to determine the best training distribution under the cost model. Because the NB algorithm is extremely time efficient, it was trained with 50:50, 40:60 and 30:70 fraud:legal distributions in the data partitions. Hence **Comparison 1:**

Which one of the above three training distributions is the best for the data partitions under the cost model?

Experiments IV and V used the best training distribution determined from Comparison 1 (either from Experiment I, II, or III) for the C4.5 and BP algorithms. The best experiment of the first three, Experiment IV and V will produce a *Bagged Z* (see Table 4) each. Experiments VI, VII, and VIII determine which ensemble mechanism produces the best cost savings. Experiment VI used bagging to combine three sets of predictions (*Bagged Z*) from each algorithm, Experiment VII used stacking to combine all predictions, and Experiment VIII proposed to bag the best classifiers determined by stacking. Experiment IX implemented the BP algorithm on unsampled and unpartitioned data, which is one of the more commonly used techniques in fraud detection commercial software. This experiment was then compared with the other six before it. Hence **Comparison 2:**

Which one of the above seven different classifier systems will attain the highest cost savings?

Experiments X, XI, and XII were constructed to find out how each sampling method performs on unpartitioned data and if they could yield better results than the multiple classifier approach. The original training data is 710 fraud examples:10627 legal examples with a 6:94 fraud:legal distribution. These three experiments were trained using C4.5 on 10:90, 20:80, 30:70, 40:60, 50:50, 60:40, and 70:30 fraud:legal distributions. Experiment X's data is obtained by undersampling the majority (legal) class to 5917, 2617, 1738, 1065, 710, 473, and 304 legal examples respectively using the top seven data partitions (see Table 7). Experiment XI's data is created by oversampling the minority (fraud) class to 1420, 2840, 4260, 7100, 10650, 15620,

and 24850 fraud examples by replicating the original 710 fraud examples. Experiment XII's data consists of the same number of examples as Experiment XI. But for Experiment XII, the minority class is innovatively oversampled by using the Synthetic Minority Oversampling TEchnique (SMOTE) [9]. This approach is more superior as it forces the decision region of the minority class to become bigger and less specific. It works by taking each specific fraud example and creating *k* artificial ones along the line segments between the randomly chosen *k* fraud class nearest neighbours. C4.5 was used as the learning algorithm for these sampling experiments. Hence **Comparison 3:**

Can the best classifier system perform better than the sampling approaches in the following results section?

4. RESULTS

Table 5 below shows in Experiments I, II, and III, the bagged success rates *X* outperformed all the averaged success rates *W* by at least 10% on evaluation sets. When applied on the score set, bagged success rates *Z* performed marginally better than the averaged success rates *Y*. Therefore, the bagged predictions were used for comparisons between all the experiments under the cost model.

Table 5: Bagged success rates versus averaged success rates

Experiment Number	*Average W*	*Bagged X*	*Average Y*	*Bagged Z*
I	71%	85%	12%	11%
II	65%	80%	67%	70%
III	68%	87%	74%	76%

Comparison 1: Experiment II achieved much higher cost savings of $94,734 compared to Experiments I (-$220,449) and III ($75,213). Therefore 40:60 fraud:legal training distribution is the most appropriate for the data partitions. Experiments IV and V were trained accordingly.

Comparison 2:

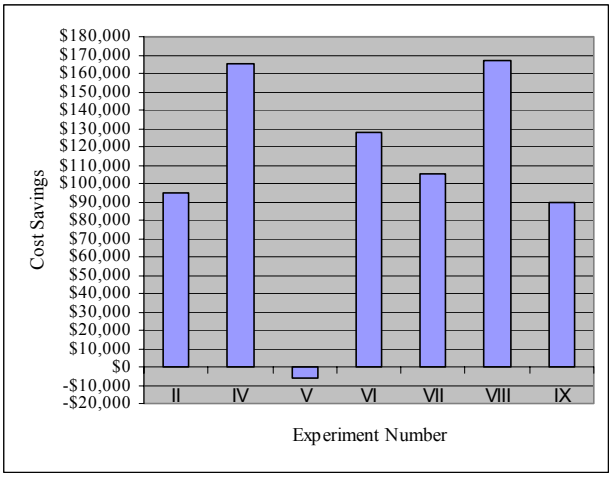

Figure 4: Cost savings results for Comparison 2

In the previous column, Figure 4 displays the cost savings of the Experiments II, IV and V which were trained, tested, and evaluated with the same eleven 40:60 data partitions. Experiment IV highlights C4.5 as the best learning algorithm for this particular automobile insurance data set with cost savings of $165,242 compared to Experiment II ($94,734) and V (-$6,488).

Figure 4 also shows the cost savings of Experiments VI, VII, and VIII, after combining NB, C4.5 and BP predictions from Experiments II, IV, and V. Both Experiment VI and Experiment VII did not produce better cost savings than the C4.5 algorithm predictions in Experiment V. However, the resultant predictions of Experiment VIII were slightly better than those of the C4.5 algorithm. Therefore, stacking-bagging creates the best multiple classifier system with the highest cost savings of $167,069. Almost all experiments (except Experiment V) outperformed the Experiment IX which was trained on the entire training data set, and scored with the best decision threshold.

Comparison 3:

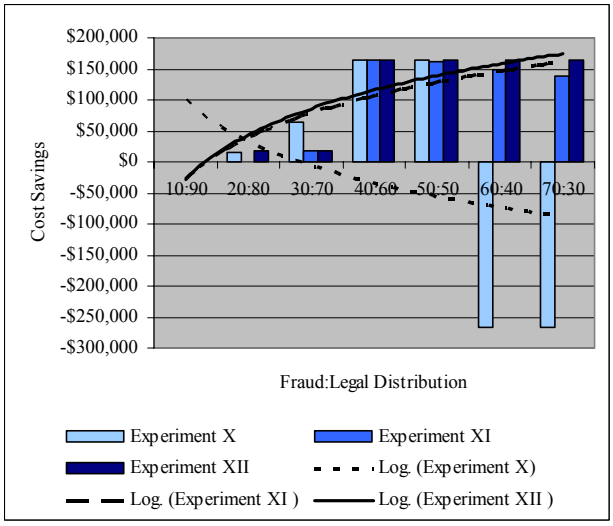

Figure 5: Cost savings results for different sampling approaches

Figure 5 above outlines the cost savings of Experiments X, XI, and XII over the different fraud:legal distributions. These three experiments performed comparably well at 40:60 and 50:50 fraud:legal distributions. Experiments XI and XII substantiate the claims that SMOTE is superior to minority oversampling with replacement, as the latter's cost savings deteriorate after 40:60. Among these three experiments, even though the undersampled data provides the highest cost savings of $165,242 at 40:60, it also incurs the highest expenditure (-$266,529) from 50:50 onwards. This is most likely due to the number of legal examples getting very small. The best multiple classifier system in the form of stacking-bagging still achieves at least $2,000 more cost savings than all the sampling variations performed here.

5. DISCUSSION

In the next page, Table 6 ranks all the experiments using cost savings. Stacking-bagging achieves the highest cost savings which is almost twice that of the conventional backpropagation procedure used by many fraud detection systems. The optimum

success rate is 60% for highest cost savings in this skewed data set and, as the success rate increases, cost savings decrease.

Table 6: Ranking of experiments using cost model

Technique (Experiment Number)	Cost Savings ($K)	Overall Success Rate	% Saved
Stacking-Bagging (VIII)	167	60%	29.7%
C4.5 40:60 Undersampled (X)	165	60%	29.4%
C4.5 40:60 Partitioned (V)	165	60%	29.4%
C4.5 40:60 SMOTEd (XII)	164	60%	29.1%
C4.5 40:60 Oversampled (XI)	164	60%	29.1%
C4.5 (2) 40:60 Partitioned	148	60%	26.4%
Bagging NB, C4.5, BP (VI)	127	64%	22.7%
Stacking All Classifiers (VII)	104	70%	18.7%
NB 40:60 Partitioned (II)	94	70%	16.9%
BP 6:94 (IX)	89	75%	15.9%
BP 40:60 Partitioned (IV)	-6	92%	-1.2%

Table 7 below illustrates the top fifteen, out of thirty three classifiers, produced from stacking. There were nine C4.5, four BP, and two NB classifiers and their predictions on the score data set were bagged. This combination of the top fifteen classifiers achieved the best predictions among the other combinations of top five, top ten, or top twenty classifiers. This supports the notion of using different algorithms with stacking-bagging for any skewed data set. It is also interesting to note that partition 1 is utilised by all the three algorithms and only partition 6 was not useful.

Table 7: Best classifiers ranked by stacking

Rank	Algorithm (Partition Number)	Rank	Algorithm (Partition Number)
1	C4.5 (2)	**9**	C4.5 (11)
2	C4.5 (4)	**10**	C4.5 (8)
3	C4.5 (1)	**11**	NB (1)
4	C4.5 (7)	**12**	BP (8)
5	NB (3)	**13**	BP (4)
6	C4.5 (5)	**14**	BP (1)
7	C4.5 (10)	**15**	BP (10)
8	C4.5 (9)		

The best NB classifier *A* and best C4.5 classifier *B*, ranked by stacking in Table 7, are compared using diversity measures such as the McNemar's hypothesis test [35] and the Q-statistic [26].

Using McNemar's hypothesis test, the null hypothesis states that both NB and C4.5 algorithms have the same success rate. On the score set, each classifier gave 3741 predictions that the other did not. 1591 correct predictions are unique to *A*, 2150 correct predictions are unique to *B*, so

$$s = \frac{\left(|1591 - 2150| - 1\right)^2}{(1591 + 2150)} = 83.2 > 6.635$$

With 99% confidence, the difference in success rates between the classifiers *A* and *B* is statistically significant.

Using the Q-statistic to assess dissimilarity of two classifier predictions, where 1 is completely similar and -1 is completely dissimilar, so

$$Q_{A,B} = \frac{(315)(27) - (2150)(1591)}{(315)(27) + (2150)(1591)} = -0.9998$$

The results from these two measures of diversity of classifier ensembles prove that stacking-bagging is robust because it chooses and combines a wide range of classifiers with very different success rates to produce the best cost savings.

6. LIMITATIONS

This paper should have used superior approaches such as SMOTE [9] to create the oversampled data partitions, and multi-response model trees [16] as the meta-classifier.

There is a fundamental limitation to the first assumption of the cost model – in reality, not all alerts will be investigated. Ranked scores with predefined thresholds are needed to direct investigations toward the instances which have the highest probability of cost savings. In fact, Pareto's law is expected to come into play: the minority of input of about 20% (reviewing the high risk claims) will produce the majority of results of about 80% (highest cost savings).

Similar to the CoIL Challenge [11] insurance data set, the number of fraudulent examples and the size of the training data set are too small. With a statistical view of prediction, 710 fraudulent training examples are too few to learn with confidence. Fraud detection systems process millions of training examples compared to a single data set with only 11337 examples. Besides that, more fraud detection data sets are also needed to increase the credibility of this paper's results.

7. FUTURE WORK

The first direction to take, as a continuation of this work, is to extend the fraud detection method based on *Minority Report* to include "analytical machinery and visual symbols" [12] to find out the properties of a data set, data partition, or data cluster which will make one classifier more appropriate than another.

Following that the next big leap forward is to work on credit application fraud detection with an industry partner. The proposed approach is to systematically examine multiple data sources such as credit application, credit bureau, white pages, and electoral data using hybrid intelligent techniques. The SOM, a form of unsupervised learning, is used for cluster detection. The ANN classifiers, a form of supervised learning, are used to generate predictions from each cluster. All the predictions are then combined by a cost-sensitive, weight-updating genetic algorithm.

8. CONCLUSION

In this paper, existing fraud detection methods are explored and a new fraud detection method is recommended. The choice of the three classification algorithms and one hybrid meta-learning technique is justified for the new method. By using this method to process the sampled data partitions, and by means of a straightforward cost model to evaluate the classifiers with, the best mix of classifiers can be picked for deployment within an organisation.

ACKNOWLEDGMENTS

We are grateful to Dr. Nitesh Chawla for providing his SMOTE algorithm and his valuable advice on running it. Our thanks also go to Mr. Rasika Amarasiri for his technical help and anonymous reviewers for their suggestions which led to more insights. This research was supported by scholarship grants to the first author from Monash University, Faculty of Information Technology, and School of Business Systems.

REFERENCES

[1] Berry M and Linoff G. *Mastering Data Mining: The Art and Science of Customer Relationship Management*, John Wiley and Sons, New York, USA, 2000.

[2] Bigus J. *Data Mining with Neural Networks*, McGraw Hill, New York, USA, 1996.

[3] Breiman L. Heuristics of Instability in Model Selection, Technical Report, Department of Statistics, University of California at Berkeley, USA, 1994.

[4] Brockett P, Xia X and Derrig R. "Using Kohonen's Self Organising Feature Map to Uncover Automobile Bodily Injury Claims Fraud", *Journal of Risk and Insurance*, USA, 1998.

[5] Cahill M, Lambert D, Pinheiro J and Sun D. "Detecting Fraud In The Real World", *in The Handbook of Massive Data Sets*, Kluwer, pp911-930, 2002.

[6] Chan P and Stolfo S. "A Comparative Evaluation of Voting and Meta-learning on Partitioned Data", *in Proceedings of 12th International Conference on Machine Learning*, California, USA, pp90-98, 1995.

[7] Chan P and Stolfo S. "Toward Scalable Learning with Non-uniform Class and Cost Distributions: A Case Study in Credit Card Fraud Detection", *In Proceedings of 4th International Conference on Knowledge Discovery and Data Mining*, New York, USA, pp164-168, 1998.

[8] Chan P, Fan W, Prodromidis A and Stolfo S. "Distributed Data Mining in Credit Card Fraud Detection", *IEEE Intelligent Systems,* **14,** pp67-74, 1999.

[9] Chawla N, Bowyer K, Hall L, and Kegelmeyer W. "SMOTE: Synthetic Minority Over-sampling TEchnique", *Journal of Artificial Intelligence Research,* Morgan Kaufmann Publishers, **16**, pp321-357, 2002.

[10] Chawla N. "C4.5 and Imbalanced Data sets: Investigating the Effect of Sampling Method, Probabilistic Estimate, and Decision Tree Structure", *in Workshop on Learning from Imbalanced Data Sets II, ICML*, Washington DC, USA, 2003.

[11] CoIL Challenge 2000. The Insurance Company Case, Technical Report 2000-09, Leiden Institute of Advanced Computer Science, Netherlands, 2000.

[12] Dick P K. *Minority Report*, Orion Publishing Group, London, Great Britain, 1956.

[13] Domingos P and Pazzani M. "Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier", *in Proceedings of the 13th Conference on Machine Learning*, Bari, Italy, pp105-112, 1996.

[14] Domingos P. "Metacost: A General Method for Making Classifiers Cost-sensitive", *In Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pp155-64, 1999.

[15] Drummond C and Holte R. "C4.5, Class Imbalance, and Cost Sensitivity: Why Under-Sampling beats Over-Sampling", *in Workshop on Learning from Imbalanced Data Sets II, ICML*, Washington DC, USA, 2003.

[16] Dzeroski S and Zenko B. "Is Combining Classifiers with Stacking Better than Selecting the Best One?", *Machine Learning*, Kluwer, **54**, pp255-273, 2004.

[17] Elkan C. Naive Bayesian Learning, Technical Report CS97-557, Department of Computer Science and Engineering, University of California, San Diego, USA, 1997.

[18] Elkan C. Magical Thinking in Data Mining: Lessons From CoIL Challenge 2000, Department of Computer Science and Engineering, University of California, San Diego, USA, 2001.

[19] Fawcett T and Provost F. "Combining Data Mining and Machine Learning for Effective User Profiling", *in Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, Oregon, USA, 1996.

[20] Fawcett T and Provost F. "Adaptive fraud detection", *Data Mining and Knowledge Discovery,* Kluwer, **1,** pp291-316, 1997.

[21] Han J and Kamber M. *Data Mining: Concepts and Techniques,* Morgan Kaufmann Publishers, San Francisco, USA, 2001.

[22] He H, Wang J, Graco W and Hawkins S. "Application of Neural Networks to Detection of Medical Fraud", *Expert Systems with Applications*, **13**, pp329-336, 1997.

[23] Insurance Information Institute. *Facts and Statistics on Auto Insurance*, NY, USA, 2003.

[24] Japkowicz N and Stephen S. The Class Imbalance Problem: A Systematic Study, *Intelligent Data Analysis*, **6**, 2002.

[25] Kalousis A, Joao G and Hilario M. "On Data and Algorithms: Understanding Inductive Performance", *Machine Learning*, Kluwer, **54**, pp275-312, 2004.

[26] Kuncheva L and Whitaker C. "Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy", *Machine Learning*, Kluwer, **51**, pp181-207, 2003.

[27] Maes S, Tuyls K, Vanschoenwinkel B and Manderick B. "Credit Card Fraud Detection Using Bayesian and Neural Networks", *in Proceedings of the 1st International NAISO Congress on Neuro Fuzzy Technologies,* Havana, Cuba, 2002.

[28] Maloof M. "Learning When Data Sets are Imbalanced and When Costs are Unequal and Unknown", *in Workshop on Learning from Imbalanced Data Sets II, ICML*, Washington DC, USA, 2003.

[29] Ormerod T, Morley N, Ball L, Langley C and Spenser C. "Using Ethnography To Design a Mass Detection Tool (MDT) For The Early Discovery of Insurance Fraud", *in Proceedings of ACM CHI Conference,* Florida, USA, 2003.

[30] Payscale. http://www.payscale.com/salary-survey/aid-8483/raname-HOURLYRATE/fid-6886, date last updated: 2003, date accessed: 28th April 2004, 2004.

[31] Prodromidis A. Management of Intelligent Learning Agents in Distributed Data Mining Systems, Unpublished PhD thesis, Columbia University, USA, 1999.

[32] Provost F. "Machine Learning from Imbalanced Data Sets 101", Invited paper, *in Workshop on Learning from Imbalanced Data Sets, AAAI*, Texas, USA, 2000.

[33] Provost F and Fawcett T. "Robust Classification Systems for Imprecise Environments", *Machine Learning*, Kluwer, **42**, pp203-231, 2001.

[34] Pyle D. *Data Preparation for Data Mining*, Morgan Kaufmann Publishers, San Francisco, USA, 1999.

[35] Salzberg S L. "On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach", *Data Mining and Knowledge Discovery*, Kluwer, **1**, pp317-327, 1997.

[36] Weatherford M. "Mining for Fraud", *IEEE Intelligent Systems*, July/August Issue, pp4-6, 2002.

[37] Williams G and Huang Z. "Mining the Knowledge Mine: The Hot Spots Methodology for Mining Large Real World Databases", *in Proceedings of the 10th Australian Joint Conference on Artificial Intelligence*, Perth, Australia, 1997.

[38] Williams G. "Evolutionary Hot Spots Data Mining: An Architecture for Exploring for Interesting Discoveries", *in Proceedings of the 3rd Pacific-Asia Conference in Knowledge Discovery and Data Mining*, Beijing, China, 1999.

[39] Witten I and Frank E. *Data Mining: Practical Machine Learning Tools and Techniques with Java*, Morgan Kauffman Publishers, California, USA, 1999.

[40] Wolpert D. "Stacked Generalization", *Neural Networks,* **5,** pp241-259, 1992.

[41] Wolpert D and Macready W. "No Free Lunch Theorems for Optimization", *IEEE Transactions on Evolutionary Computation*, **1**, pp67-82, 1997.