Decision Support Systems 50 (2011) 576–584

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

An ANN-based auditor decision support system using Benford's law

Sukanto Bhattacharya [a,*], Dongming Xu [b], Kuldeep Kumar [c]

[a] Deakin University, Australia
[b] University of Queensland, Australia
[c] Bond University, Australia

ARTICLE INFO

Available online 18 August 2010

Keywords:
Benford's law
Auditor decision support system
ARPs
ANNs
Genetic optimization

ABSTRACT

While there is a growing professional interest on the application of Benford's law and "digit analysis" in financial fraud detection, there has been relatively little academic research to demonstrate its efficacy as a decision support tool in the context of an analytical review procedure pertaining to a financial audit. We conduct a numerical study using a genetically optimized artificial neural network. Building on an earlier work by others of a similar nature, we assess the benefits of Benford's law as a useful classifier in segregating naturally occurring (i.e. non-concocted) numbers from those that are made up. Alongside the frequency of the first and second significant digits and their mean and standard deviation, a posited set of 'non-digit' input variables categorized as "information theoretic", "distance-based" and "goodness-of-fit" measures, help to minimize the critical classification errors that can lead to an audit failure. We come up with the optimal network structure for every instance corresponding to a 3×3 *Manipulation–Involvement matrix* that is drawn to depict the different combinations of the level of sophistication in data manipulation by the perpetrators of a financial fraud and also the extent of collusive involvement.

1. Introduction

Analytical review procedure (ARP) is one of a number of tools in an external auditor's toolbox to ascertain the credibility of an organization's financial reports. As per SAS 99 auditors are now expected to collect and consider a much more information than they did in the past in order to better assess fraud risks [36]. However fraud detection is still not universally perceived as being the primary responsibility of an external auditor. There exists a serious "expectation gap" between what various stakeholders perceive as auditor's primary responsibilities and what auditors are capable of doing or are equipped to do given time and budget constraints [45]. Of course; if a fraud has been committed chances are that some numbers would appear 'out of place' to an auditor, thus causing "red flags" to be raised and warranting a deeper investigation. ARPs are specialized *auditor decision support systems* intended to make the audit process more efficient by quickly identifying the 'out of place' numbers. ARPs have been attributed with the ability to detect more anomalies than they are typically given credit for if other detection procedures had failed [44].

However some degree of subjectivity is involved in traditional ARPs which rely heavily on the auditors' judgment. Our main research objective here is to try and build on earlier research on an ANN-based ARP that applies a particularly useful statistical law known as Benford's law and is less prone to subjective factors. Towards this objective, we firstly review relevant literature in Section 2. Secondly, in Section 3, we propose an ANN-based auditor decision support system to re-examine the efficacy of Benford's law in helping to correctly discriminate between data sets that are naturally occurring and others which aren't. Thirdly, in Section 4, we specify a proposed ANN-based auditor decision support system with an aim to improve on the classification results obtained by earlier researchers by identifying and testing input variables to minimize the number of critical errors. Finally, in Sections 5 and 6, we analyse our system output, compare it with the results of earlier researchers, draw conclusions and identify limitations.

2. Literature review

2.1. Artificial intelligence-based ARPs

Coakley and Brown [10] sought to improve a financial ratio analysis-based ARP by applying a neural network to select the most optimal ratios for the task. Fanning et al. [16] and Fanning and Cogger [15] firstly posited an ANN-based approach to detect managerial fraud and then applied it in a later paper using published financial data. Green and Choi [20] attempted something similar to Fanning and Cogger's paper a year later. Welch, Reeves and Welch [43] developed a classifier system for modeling auditor decision behavior in a fraud setting by applying a genetic algorithm approach. Feroz et al. [17] applied ANNs to study the efficacy of the "red flags" approach. Coakley and Brown [11] addressed some of the deeper modeling issues with

* Corresponding author. Tel.: +61 3 9244 6544.
E-mail address: sukanto.bhattacharya@deakin.edu.au (S. Bhattacharya).

0167-9236/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.dss.2010.08.011

ANN applications in accounting and finance. Lin et al. [30] further extended the ANN as a methodological tool in ARP by applying a fuzzy neural network model in the assessment of risk pertaining to fraudulent financial reporting. Kirkos et al. [25] tested the usefulness of decision trees, neural networks and Bayesian belief networks in identifying fraudulent financial reporting using ratios derived from financial statements to construct the input vectors.

However most previously published research works relied primarily on a single backbone AI technology rather than combining two or more different ones to obtain a more optimal classification engine for the specific task at hand. We on the other hand have attempted a hybrid approach by using a genetically optimized neural network, which ensures the most optimal neural network configuration in terms of both its architecture and input variable selection. Since the governing optimization problem is fairly complex and unstructured, Genetic Algorithm (GA) is likely the best approach to optimize the ANN and obtain best prediction results with test set data. A GA optimizer also makes the network building phase more efficient by rendering comparative analyses of alternative configurations superfluous, because by the principle of natural selection, only the most optimal configuration is expected to survive the evolutionary optimization process.

2.2. Benford's law in fraud detection

Frank Benford [2] collected numerical data on a wide variety of subjects in support of an observation that although the numbers were randomly selected, their digits followed a certain probability distribution that was not quite in accordance with human intuition (intuitively the digits should have an equal probability of occurrence). For the first significant digit, the formula for this probability distribution is as follows:

$$P(D) = \log_e[(1 + 1/D)] / \log_e(10) \quad (1)$$

In the formula D is any digit from 1 to 9. This formula can of course be generalized to cover second and subsequent significant digits. However the difference in probability of occurrence is most striking for the first significant digit. And beyond the second significant position, the occurrence probabilities tend to approximately converge to their intuitive equal values. Benford's law is analogous to another observation on rank-order occurrence that goes by the name of Zipf's law in honor of its discoverer [46]. Mathematically, Benford's law may actually be a special case of Zipf's Law [35]. The defining property of Benford's law is that it is observable only in *naturally occurring* numbers—not in numbers that have been artificially concocted. Thus, Benford's law may be regarded as a veritable *signature of Nature*—something that cannot be replicated manually [28]. This is the precise property that makes Benford's law extremely useful in detecting fraudulent financial data. When it comes to fraudulent manipulation of financial accounts, concocted numbers will not obey Benford's law thus increasing chances of detection via ARPs based on this law. Under a standard double-entry book-keeping system, figures that are made up to essentially "plug the gaps" caused by fraud can result in differences in the observed 1st and 2nd-digit frequencies from those predicted by Benford's law. Benford's law has very useful mathematical properties of base and scale invariance. Therefore an ARP that uses Benford's law is not affected by magnitude or 'history' of a transaction and therefore can be very effective in detecting "bleeding frauds" where small amounts are fraudulently siphoned over a period of time off via dubious transactions without alerting internal controls. Traditional review procedures while being able to identify frauds that involve a single high-value transaction may not however be effective in detecting such frauds. However, simply a failure to comply with Benford's law does not necessarily imply fraud—it merely provides some statistical evidence that the data may have been manipulated but does not reveal whether such manipulation is fraudulent or benign. So while Benford's law may be a useful tool, it has certain limitations which have been discussed by Kumar and Bhattacharya [28].

The background and development of Benford's law, as an effective tool in forensic accounting, has recently been comprehensively reviewed [14]. Significant academic research, albeit isolated, on the first digit law and its applications in financial fraud detection had been done previously. Carslaw [8] did an exploratory study concerning detection of anomalies in income numbers. Soon after, Thomas [41] used Benford's law to detect unusual patterns in reported earnings. But it was Mark Nigrini [31–33] who virtually pioneered application of Benford's law in forensic accounting by applying it to cases of tax evasion and other types of financial fraud. Dalal [12] reported a case where forensic accountants successfully used Benford's law in a real-life investigation to identify dubious transactions in one of the biggest financial frauds of recent times, which ultimately led to the collapse of a major international bank. Kumar and Bhattacharya [27] sought to design a computational algorithm based on Benford's law with the aim of making an audit sampling process more efficient. Watrin et al. [42] found under an experimental setting that subjects did not adapt to Benford's law while concocting numbers thereby lending credence to the robustness of this technique as an effective detection mechanism. S.-M. Huang et al. [22] proposed an innovative fraud detection mechanism based on Zipf's law. However the *only* paper to date published in a peer-reviewed, academic journal that combines Benford's law and ANNs as a review procedure is the one published in the Managerial Auditing Journal by Bruce Busta and Randy Weinberg [7], which largely motivated our current work.

2.3. The Manipulation–Involvement hypothesis

When a perpetrator of a fraud is highly resourceful, knowledgeable and organized, it is reasonable to assume that the manipulation of transaction records or books of accounts would have been done with a fair amount of sophistication so as to evade detection by any of the internal control systems in place. Moreover, when there is collusion it is also likely that a significant percentage of the detectable manipulations at the principal point of origin of the fraud would get suppressed as the perpetrators cooperate to 'cover up' their tracks at their respective ends effectively negating most if not all the 'check-and-balance' type internal control systems that are commonly used. Intuitively therefore, the complexity of a financial fraud (i.e., how well it is concealed so as to evade detection by reasonably alert internal control systems) would depend on both the level of sophistication in the manipulation of the financial records (the "Manipulation" variable) and the extent of involvement of multiple perpetrators (the "Involvement" variable). This is basically in line with the

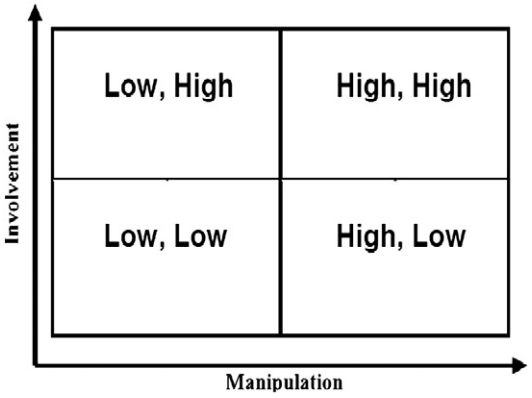

Fig. 1. A schema of the manipulation and involvement variables in collusive financial frauds.

Table 1
First and second digit frequencies for Benford distribution.

Digit	Benford prob. of being the 1st sig. digit	Benford prob. of being the 2nd sig. digit
0	–	0.11968
1	0.30103	0.11389
2	0.17609	0.10882
3	0.12494	0.10433
4	0.09691	0.10031
5	0.07918	0.09668
6	0.06695	0.09337
7	0.05799	0.09035
8	0.05115	0.08757
9	0.04576	0.08500

Manipulation–Involvement (hereafter *M–I*) originally posited by Bhattacharya et al. in 2005 [4]. Fig. 1, adopted from [3], illustrates this concept:

Intuitively, the {low manipulation, low involvement} rectangle would be expected to contain the typically easy-to-detect cases of fraud, whereas the {high manipulation, high involvement} rectangle would be expected to have the typically more difficult-to-detect ones [3].

2.4. Getting the numbers

Hill [21] ran an experiment and obtained a set of concocted numbers that did not conform to Benford's law. Busta and Weinberg (hereafter B&W) assumed that any concocted data will resemble the Hill distribution because the underlying cognitive generating process would be similar [7]. We agree with this and like B&W, have used the Hill numbers in our work. Tables 1 and 2 depict the probability distributions of the first and second significant digits for Benford and Hill numbers respectively [7].

3. ANN-based binary decision support system for ARP

A binary data classification system is designed to signal which one of two pre-specified groups a data point belongs to. The two groups in our system are coded as "Benford" (0) and "non-Benford" (1) and the data classification system will classify a data point either as "0" (implying it has come from a Benford distribution) or "1" (implying that it has come from a non-Benford distribution). Where the signal is "0" but the data set does not follow Benford's law, and then there may be some degree of 'contamination' in it that has not been picked by the decision support system. This is a "Failed Alarm"; and can potentially cause audit failure as pointed out by B&W [7]. Where the signal is "1" but the data set really is naturally occurring and obeys Benford's law; it is a case of a "False Alarm"; and this can have potential time-cost impacts due to *over-auditing*. Obviously with any practical ARP, "Failed Alarm" errors carry a far more serious and potentially hazardous consequence compared to "False Alarm" errors Fig. 2. While we concede that this nomenclature of these two statistical equivalent of Type I and Type II errors is somewhat unconventional, we believe that it fits the scheme of things really well in terms of making clear the nature of these errors to any uninitiated reader and helping to effectively distinguish between the two types of possible classification error. As is perhaps quite intuitive, there is a trade-off between the "Failed Alarm" and "False Alarm" errors and the emphasis in our ARP decision support system is mainly on the minimization of "Failed Alarm" error as it is critical. However "False Alarm" is also considered.

Table 2
First and second digit frequencies for Hill distribution.

Digit	Hill prob. of being the 1st sig. digit	Hill prob. of being the 2nd sig. digit
0	–	0.05800
1	0.14700	0.10600
2	0.10000	0.11700
3	0.10400	0.10900
4	0.13300	0.10500
5	0.09700	0.10000
6	0.15700	0.11200
7	0.12000	0.12800
8	0.08400	0.07300
9	0.05800	0.09200

Our neural networks are the fully connected; backpropagation type and were constructed using the Neuralyst™ v1.4 software which is produced and marketed by California-based Cheshire Engineering Corporation and runs as an add-on to Microsoft Excel. The *sigmoid transfer function* is used to map the output to the binary form. Further details about neural network particulars and the Neuralyst™ v1.4 software are supplied in Appendix I. We ran Neuralyst™'s built-in GA optimizer to select the most optimal network configuration for minimizing the *root-mean-square* (RMS) training error, which was the default fitness criterion. The GA optimizer performed both structure optimization as well as parameter optimization and some of the operational details are further explained in Appendix II. While we started off with a three-layer ANN configuration having a single hidden layer in all the nine data groups, the GA optimizer retained the three-layer configuration in only two of them; and gave a four-layer configuration to the remaining seven.

4. The system setup

4.1. The input variables

4.1.1. The "digit" variables

Following B&W [7], we chose to use the *observed first* and *second digit frequencies* in the contaminated and non-contaminated data sets as input variables for training the ANN. This meant a total of 19 input variables that were "digit" variables—9 corresponding to the first digit frequencies and 10 corresponding to the second digit ones (as this included the digit 0).

4.1.2. The descriptive statistics

Again following B&W [7], we chose to include the *mean* and the *standard* deviation of the observed first and second digits but decided to leave out median, skewness and kurtosis to avoid a clutter of input variables of the descriptive type. This meant a total of 4 variables of this type—for the first and the second digits. The mean of the first significant digits was calculated as under:

$$m_{1st} = \Sigma p_i D_i, i = 1, 2, \ldots, 9 \quad (2)$$

Here D_i is the ith digit and p_i is its observed probability of occurrence as the first significant digit in each data set (contaminated and non-contaminated). Similarly, the mean of the second significant digits was calculated as under:

$$m_{2nd} = \Sigma p_i D_i, i = 0, 1, 2, \ldots, 9 \quad (3)$$

The standard deviation of the first significant digits was calculated as under:

$$s_{1st} = \sqrt{\left[\Sigma p_i (D_i - m_{1st})^2\right]}, i = 1, 2, \ldots, 9 \quad (4)$$

	Data obeys Benford's law	*Data does not obey Benford's law*
System response is "0"	*No error*	*Failed Alarm*
System response is "1"	*False Alarm*	*No error*

Fig. 2. Error classification for the binary response ARP using Benford's law.

Similarly, the standard deviation of the second digits was calculated as under:

$$s_{2nd} = \sqrt{\left[\Sigma p_i (D_i - m_{2nd})^2\right]}, i = 0, 1, 2, \ldots, 9 \quad (5)$$

4.1.3. Additional non-digit variables

Our approach differs from B&W due to the inclusion of a number of other non-digit variables besides the descriptive statistics. The digit variables, as B&W had concluded [7], helped in better identifying the Benford data—i.e. they were purported to be better suited in reducing the "False Alarm" type of classification errors. Our system uses additional non-digit input variables, *which are selected so as to help the ANN to better learn the underlying pattern in the numbers in order to correctly identify the non-Benford data*—i.e. they are purported to be better suited in reducing the more critical "Failed Alarm" type of classification errors. Our non-digit input variables fall under three main categories—goodness-of-fit statistics, information theoretic measures and distance-based measures. *Each of these measures helps to reveal the statistical disparity between numbers drawn from Benford and non-Benford distributions. So adding these non-digit variables makes our system more potent in avoiding "Failed Alarms" that are more critical from an audit perspective*.

4.1.4. The goodness-of-fit statistics

Our system includes two goodness-of-fit statistics—the *chi-squared* and the *discrete Kolmogorov–Smirnov* statistic [34]. The chi-squared statistic normally is quite reliable in goodness-of-fit tests and has been widely applied in testing whether a set of data originates from a Benford distribution. In Steele and Chaseling [39] however it was shown that for trending distributions (like the Benford one) the discrete Kolmogorov–Smirnov statistic yields higher power in a goodness-of-fit test compared to the chi-squared. We decided to use both chi-squared as well as the Kolmogorov–Smirnov statistics to provide as much variations as possible in the input data in order to enable the ANN to quickly learn the underlying distinctive statistical patterns.

The chi-squared statistic has been calculated as under for the first digits:

$$\chi^2_{1st} = \Sigma (f_{1i} - B_{1i})^2 / B_{1i}, i = 1, 2, \ldots, 9 \quad (6)$$

Here f_{1i} is the observed first digit frequency for the ith digit and B_{1i} is the expected frequency of the ith digit as the first digit as per the Benford distribution. The chi-squared statistic has been calculated as under for the second digits:

$$\chi^2_{2nd} = \Sigma (f_{2i} - B_{2i})^2 / B_{2i}, i = 0, 1, 2, \ldots, 9 \quad (7)$$

Here f_{2i} is the observed second digit frequency for the ith digit and B_{2i} is the expected frequency of the ith digit as the second digit as per Benford distribution.

The discrete Kolmogorov–Smirnov statistic has been calculated as under for the first digits:

$$K_{1st} = \max_i \left| \Sigma \left(f_{1j} - B_{1j} \right) \right|, i, j = 1, 2, \ldots, 9 \quad (8)$$

Again f_{1j} is the observed first digit frequency for the jth digit and B_{1j} is the expected frequency of the jth digit as the first digit as per the Benford distribution. Similarly, the discrete Kolmogorov–Smirnov statistic has been calculated as under for the second digits:

$$K_{2nd} = \max_i \left| \Sigma \left(f_{2j} - B_{2j} \right) \right|, i, j = 0, 1, 2, \ldots, 9 \quad (9)$$

Again f_{2j} is the observed second digit frequency for the jth digit and B_{2j} is the expected frequency of the jth digit as the second digit as per Benford distribution. So; again; there are 4 of this type of variables—two each corresponding to the first and second significant digits.

4.1.5. The information theoretic measures

We used two information theoretic measures in our data classification system—the *modified Kullback–Leibler distance* and the *cumulative Shannon entropy differential*. Each of these measures helps to identify the 'divergence' between two probability distributions and would as such help the ANN in distinguishing the numbers drawn from a non-Benford distribution. Again, there are a total of 4 of information theoretic input variables—two each for the first and second digits. The modified Kullback–Leibler distance as proposed by Kim and Cho [24], is a subsequent improvement over the original Kullback–Leibler distance [26] and measures the difference between two discrete probability distributions p and q as follows:

$$D(q, p) = D(p, q) = (0.5)\Sigma_k (p_k \log p_k / q_k + q_k \log q_k / p_k) \quad (10)$$

Here p_k corresponds to the probability of occurrence of the kth digit as the first significant digit in a Benford distribution, while q_k corresponds to the probability of occurrence of the kth digit as the first significant digit in a non-Benford distribution and k = 1, 2, …, 9 for the modified Kullback–Leibler measure corresponding to the distribution of the first significant digits. Similarly, p_k corresponds to the probability of occurrence of the kth digit as the second significant digit in a Benford distribution, while q_k corresponds to the probability of occurrence of the kth digit as the second significant digit in a non-Benford distribution and k = 0, 1, 2, …, 9 for the modified Kullback–Leibler measure corresponding to the distribution of the second significant digits.

The cumulative Shannon entropy differential measures the cumulative absolute difference in the *expected information content* [37] or entropy of a distribution that obeys Benford's law from one that doesn't. The measure has been calculated as under:

$$\delta(p, q) = \Sigma_k |\Sigma p_k \log_2 p_k - \Sigma q_k \log_2 q_k| \quad (11)$$

Here p_k corresponds to the probability of occurrence of the kth digit as the first significant digit in a Benford distribution, while q_k corresponds to the probability of occurrence of the kth digit as the first significant digit in a non-Benford distribution and k = 1, 2, …, 9 for the cumulative Shannon entropy differential corresponding to the distribution of the first significant digits. Similarly, p_k corresponds to the probability of occurrence of the kth digit as the second significant digit in a Benford distribution, while q_k corresponds to the probability of occurrence of the kth digit as the second significant digit in a non-Benford distribution and k = 0, 1, 2, …, 9 for the cumulative Shannon

entropy differential corresponding to the distribution of the second significant digits.

4.1.6. The "distance"-based variables

We have included three "distance" measures in input variables. The first and the simplest is the *Pearson's coefficient of correlation* $\rho_{p,q}$ between the expected and observed frequencies of the first digits (and second digits as well) with those expected as per Benford's law. The measure is calculated as under:

$$\rho_{p,q} = \mathrm{covar}(p,q) / \left(s_p\right)\left(s_q\right) \tag{12}$$

For the coefficient of correlation between the expected and observed frequencies of the first digits, covar(p, q) corresponds to the covariance of expected and observed first digit frequencies, while s_p and s_q are standard deviations of the expected and observed first digit frequencies. Similarly, for the coefficient of correlation between the expected and observed frequencies of the second digits, covar(p, q) corresponds to the covariance of expected and observed second digit frequencies, while s_p and s_q are standard deviations of the expected and observed second digit frequencies.

The second distance-based measure included as an input variable is the *Euclidean distance* measure which has also been previously applied in the context of determining how closely a data set follows the Benford distribution and is calculated as under [9]:

$$\varepsilon = \left[\sqrt{\Sigma_k (p_k - q_k)^2}\right] / \left[\sqrt{\left\{\left(\Sigma_{k-1} p_k^2\right) + (p_9 - q_9)^2\right\}}\right] \tag{13}$$

Again; p_k and q_k correspond to the expected (by Benford's law) and observed first digit frequencies and $k = 1, 2, \ldots 9$ for the Euclidean distance measure corresponding to first digits and p_k and q_k correspond to the expected and observed second digit frequencies and $k = 0, 1, 2, \ldots 9$ for the Euclidean distance measure corresponding to second digits.

The third and last distance-based measure included as an input variable is the Judge–Schechter *alpha*, which is the absolute value of the difference between the average of the observed distribution and the average of the expected distribution divided by the maximum possible difference [23].

So we have a total of 37 input variables—19 digit variables and the rest non-digit variables.

4.2. Constructing the data sets

We have constructed 800 separate data sets, each containing 1000 two-digit numbers (B&W's data sets consisted of only 200 such numbers [7]). These data sets have been generated by Monte Carlo simulation on a customized MS Excel spreadsheet. In addition to a non-manipulated type of data set constructed purely from the Benford distribution, we constructed three 'manipulated' types—each having three levels of detectable manipulation—at 10% (indicating a "high level of sophistication", 20% (indicating a "moderate level of sophistication") and 50% (indicating a "low level of sophistication). This means that 90%, 80% and 50% of these data sets respectively are made up of Benford numbers and the balance are made up of numbers generated via a Hill distribution. Three detectable involvement levels are also postulated—"high", "moderate" and "low". For the "high" involvement level, 720 data sets correspond to the non-manipulated type and 80 data sets correspond to the manipulated type for each level of detectable manipulation. For the "moderate" involvement level, 560 data sets correspond to the non-manipulated type and 240 correspond to the manipulated type for each level of detectable manipulation. For the "low" involvement level 400 data sets correspond to the non-manipulated type and 400 data sets are manipulated for each level of detectable manipulation. Our computational setup is shown below (Fig. 3):

Involvement \ Manipulation	Low	Moderate	High
High	H-L	H-M	H-H
Moderate	M-L	M-M	M-H
Low	L-L	L-M	L-H

Fig. 3. Computational setup.

The logic that has been applied to make the data conform to this setup basically goes like this: if there is a high level of sophistication in concealing the fraudulent transactions, a relatively smaller fraction of the dubious numbers would be detectable in the financial records as compared to a lower level of sophistication. In terms of the involvement of multiple perpetrators, if there is a high level of collusive involvement present, a small percentage of the contaminated records are expected to actually come under independent scrutiny as compared to a case of a lower level of collusive involvement (Tables 3).

It must however be stressed that our selected contamination percentage cut-offs for the various groups corresponding to low, moderate or high degrees of manipulation/involvement are wholly subjective. We might as well have chosen 95%–5% for high, 75%–25% for moderate and 55%–45% for low—or any other such combination as

Table 3
Distribution of contaminated and non-contaminated data among the nine groups.

Involvement	Manipulation: Low	Moderate	High
High	Pure Benford—720 Benford-Hill (50%–50%)—80	Pure Benford—720 Benford-Hill (80%–20%)—80	Pure Benford—720 Benford-Hill (90%–10%)—80
Moderate	Pure Benford—560 Benford-Hill (50%–50%)—240	Pure Benford—560 Benford-Hill (80%–20%)—240	Pure Benford—560 Benford-Hill (90%–10%)—240
Low	Pure Benford—400 Benford-Hill (50%–50%)—400	Pure Benford—400 Benford-Hill (80%–20%)—400	Pure Benford—400 Benford-Hill (90%–10%)—400

Table 4
Model summary and prediction results for the H-H group using binary logistic regression.

Model Summary	−2 Log likelihood 190.3462	Cox and Snell R^2 0.3378	Nagelkerke R^2 0.7067
Observed Y	Predicted Y: 0	1	Percentage correct
0	705	15	97.92%
1	27	53	66.25%

Table 5
Significant explanatory variables in the binary logistic regression equation.

		B	S.E.	Wald	Df	Sig.
Step 1	FD6	1.675	1.012	2.737	1	0.098*
	FD7	1.357	0.759	3.200	1	0.074*
	FD8	0.722	0.426	2.870	1	0.09*
	KS_Stat_FD	−0.150	0.075	4.056	1	0.044**
	Pearson_r_FD	762.49	244.770	9.704	1	0.002***
	KL_FD	2083.86	1124.051	3.437	1	0.064*
	Pearson_r_SD	−19.737	11.087	3.169	1	0.075*

*** Significant at 1% level.
** Significant at 5 level.
* Significant at 10% level.

Table 6
Network training statistics and parameters for the H-H group for a 3-layer baseline ANN.

Training RMS Error	0.2500
Number of training iterations	1500
Learning rate	1.0
Momentum	0.9
Input noise	0.0
Cut-off "False Alarm"	0.20>
Cut-off "Failed Alarm"	0.80<
Number of layers	3
Number of hidden layers	1
Number of input neurons	37
Number of hidden neurons	1
Number of output neurons	1

long as our choice adequately reflected the underlying logic of the *M–I* hypothesis that we are trying to computationally apply in this posited ANN-based decision support system for ARP.

4.3. Numerical output and validation

In a number of fraud detection-related empirical works done previously, logistic regression was used as the governing methodology. Notable ones among these works include a study of the relation between the board of director composition and financial statement fraud [1], a study of the relation between fraud type and auditor litigation [6] and another that threw light on the link between earnings and operating cash flows and incidence of financial reporting fraud [29]. However artificial neural networks (ANNs) was chosen as our primary tool as they are a nonlinear, nonparametric function estimation technique that tends to perform better in fitting general nonlinear, multivariate functions in situations where *multicollinearity* is present. Due to the inter-related nature of many of our input variables they are expected to be highly correlated, which could potentially lead to problems of multicollinearity. We in fact did run a full set of *binary logistic regression models* using SPSS (results only for the H-H group reported for the sake of brevity) with same input data as the ANN models (Tables 4 and 5). While the prediction results compare favourably with that obtained using GA-optimized ANNs, only 2 out of the 37 explanatory variables came out as significant at the 5% level possibly due to the presence of a high degree of multicollinearity.

To gauge the comparative performance of binary logistic regression vis-à-vis a simple ANN classification model, we have set up a *baseline* ANN classifier with the simplest possible 3-layer architecture (one input layer, one hidden layer and one output layer) and have run it with the same input data for the H-H group. This baseline ANN has the simplest possible 3-layer architecture with only one hidden layer containing a single neuron; as a simpler configuration would mean that the ANN would have to have a 2-layer architecture in which case the ANN model would become quite akin to a regression model and would lose much of its characteristic pattern learning power. The results shown do provide a comparative insight vis-à-vis the logistic regression model as well as the GA-optimized ANNs that we present later in this section. The network parameters in the baseline ANN are same as those configured in Neuralyst™ by default. The maximum number of training iterations is kept at 1500 in order to avoid any loss in predictive utility due to overtraining (Table 6).

While the binary logistic regression model shows a 97.92% in-sample prediction accuracy against "False Alarm" errors (compared to 99.72% for the baseline ANN) and a 66.25% in-sample prediction accuracy against "Failed Alarm" errors (compared to 40% for the baseline ANN), none of the digit variables are significant at 5% level while only Pearson's correlation coefficient for 1st significant digit and the Kolmogorov–Smirnov statistic for 1st significant digit come out as significant at the 5% level out of the non-digit variables (Table 7). However, a number of the other explanatory variables do contribute non-negligible information as we can see from our subsequent results with the GA-optimized ANNs presented later in this section. The logistic regression model fails to capture the significance of many informationally valuable variables due to the likely presence of a high degree of multicollinearity given the inter-related nature of these variables.

Table 7
Comparative prediction performances for the II–II data group.

	"False Alarm" errors	"Failed Alarm" errors
Logistic regression	2.08%	33.75%
Baseline ANN	0.28%	60%

Table 9
Genetic training parameters.

Generation count	10
Structure count	3
Least epochs	100
Inclusion rate	0.75
Max layers	4
L2 neuron limit	30
L3 neuron limit	10
L4 neuron limit	0
L5 neuron limit	0
Min learning rate	0.5
Max momentum	1
Max input noise	0.03
Population size	3
Population mode	Immigrate
Crossovers	1
Mutation rate	0.1
Fitness criteria	Train error
Fitness limit	100

Note: The GA optimizer parameters are same for all nine groups, so we have reported it only for the first group. The selected parameters are same as the default settings for Neuralyst v1.4. One could of course tweak any of these around a bit but it would be a rather time-consuming exercise and at the end; is unlikely to have much of an impact on the network prediction performances.

Table 8
Network training statistics, parameters and configuration for the nine groups.

	Training RMS error	No. of training iterations	Learning rate	Momentum	Input none	Cut-off "false Alarm"	Cut-off "Failed Alarm"	Layers	Input neurons	Hidden neurons—1st layer	Hidden neurons—2nd layer
L-L	0.0000	46	0.845958	0.547136	0.008256	0.20>	0.80<	4	32	23	3
M-L	0.0000	71	0.527573	0.154576	0.021850	0.20>	0.80<	4	28	27	1
H-L	0.0000	107	0.943952	0.489456	0.019348	0.20>	0.80<	4	28	15	5
L-M	0.0000	989	0.810266	0.170019	0.019236	0.20>	0.80<	4	29	25	5
M-M	0.0707	1500	0.974105	0.386792	0.024932	0.20>	0.80<	4	30	3	10
H-M	0.0500	1500	0.511567	0.468795	0.009015	0.20>	0.80<	4	28	22	5
L-H	0.2761	1500	0.651723	0.425306	0.021760	0.20>	0.80<	3	29	18	–
M-H	0.2872	1500	0.720557	0.603870	0.006192	0.20>	0.80<	3	29	7	–
H-H	0.1581	1500	0.728019	0.028474	0.010606	0.20>	0.80<	4	27	6	5

Table 10
ANN performance in terms of "Failed Alarm" error for various *M–I* levels.

	Manipulation		
Involvement	Low	Moderate	High
High	Train—0% Test—0%	Train—2.5% Test—20%	Train—23.75% Test—80%
Moderate	Train—0% Test—0%	Train—167% Test—0%	Train—15.42% Test—20%
Low	Train—0% Test—0%	Train—0% Test—0%	Train—0.75% Test—20%

Table 11
ANN performance in terms of "False Alarm" error for various *M–I* levels.

	Manipulation		
Involvement	Low	Moderate	High
high	Train—0% Test—0%	Train—0% Test—0%	Train—0.14% Test—0%
Moderate	Train—0% Test—0%	Train—0% Test—0%	Train—5.18% Test—20%
Low	Train—0% Test—0%	Train—0% Test—0%	Train—14.50% Test—20%

For the posited GA-optimized ANN-based decision support system for ARP, we present the network statistics (Tables 8 and 9) and prediction performance (Tables 10 and 11) for all nine groups in training as well as test sets, which are all of equal sizes for all the nine groups and each test set is made up of 50% Benford data sets and 50% Hill data sets at varying levels of manipulation (at 10%, 20% and 50%).

5. Analysis and discussion

5.1. Comparison with B&W's results

The prediction performance in terms of "Failed Alarm" error (i.e. failure to classify a non-Benford data set as such) is best for a low level of sophistication in manipulation (0% error), which totally agrees with B&W's results as they too had obtained a 100% correct prediction for a 50% level of contamination in data [7]. For a 10% level of contamination, B&W could only get a maximum of 73.20% correct prediction ("Failed Alarm" error of 26.80%) with their best-case ANN design [7]. We have a 76.25% correct training set prediction ("Failed Alarm" error of 23.75%) corresponding to the hardest-to-detect H-H group that has only 10% of the 800 data sets manipulated by data contaminated to a 10% level. This training set prediction performance compares very favourably also against our baseline ANN results as well as the logistic regression results reported earlier for this particular data group. For the same level of sophistication in manipulation (i.e. 10% detectable contamination in data) but a moderate level of involvement, we have 84.58% correct training set prediction. It further improves to a 99.25% correct training set prediction for low involvement. B&W had also concluded that "… the digital frequencies are important in the identification of uncontaminated data" [7]. We do find in our results that the GA optimizer has retained most of the 19 "digit" variables for a low involvement level across all three levels of sophistication in manipulation (19, 15 and 14 "digit" variables retained respectively). This observation is also seen to hold true for a moderate involvement level across all three levels of sophistication in manipulation (17, 16 and 14 "digit" variables retained respectively). But it is not seen to hold for a high involvement level across all three levels of sophistication in manipulation (13, 13 and 14 "digit" variables retained respectively). So while we do find some support for B&W's conclusion on the role of the digit variables, we must say that this support is at best of a weak nature [7].

5.2. 'Fitness' of the non-digit variables

Unlike regression analysis, a neural network system cannot yield statistical measures of significance corresponding to the input variables used. However, the GA optimizer effectively performs similar function as a *backward stepwise algorithm* in case of a multiple regression model. We chose not to consider the moderate and low manipulation cases because the ANN faced stiffest learning difficulties for the three cases of high manipulation. Overall the information theoretic and goodness-of-fit statistics appear "fittest" among the non-digit variables. The input variables that were not retained in the final network for each of the nine groups are tabulated below. These are the ones that failed to survive the evolutionary optimization process and so were eliminated (Table 12).

5.3. Tying in with the M–I hypothesis

Our numerical results clearly tie in with the *M–I* hypothesis originally postulated by Bhattacharya et al. [4]. In general, the ANN prediction performances are substantially better for low and moderate level of sophistication in manipulation as compared to a high level of sophistication in manipulation across all the three levels of involvement. This goes along with intuitive logic as it is expected that no matter what the level of involvement is, a certain minimum level of sophistication in executing the fraud will be required to get the manipulations past the internal controls.

6. Conclusion—significance, shortcomings and future directions

In this paper, we used the *Manipulation–Involvement* hypothesis to build and test a ANN-based system for binary data set classification based on whether or not a data set conforms to Benford's law. This we contend could be a very useful tool in the form of an ARP as was previously proposed [7]. Our study, being based on the *M–I* hypothesis, likely possesses a more solid theoretical foundation than the prior examples; thereby making what we believe is a significant contribution to existing body of knowledge on quantitative methods for financial fraud detection.

In addition to the first and second significant digit frequencies and related descriptive statistics, our posited system additionally uses a number of non-digit input variables and results indicate that this decision support system has fewer "Failed Alarm" errors. Of particular interest is the relatively higher level of 'fitness' with respect to reducing "Failed Alarm" training error displayed by the cumulative entropy differential and the discrete Kolmogorov–Smirnov goodness-of-fit statistic. This warrants further research in the future to identify

Table 12
Input variables *not included* in the GA-optimized—ANN.

	Manipulation		
Involvement	Low	Moderate	High
High	1, 5, 6, Chi-stat (1), K-L distance (1), 0 7, 9	1, 5, 9, Mean (1), KS-stat (1), 0, 2, 6	1, 2, 8, 9, Mean (1). J-S alpha (1), 4, Pearson's 1(2), J-S alpha (2). Euclidean distance (2)
Moderate	7, S.D.(1), 0. Mean (2), SD.(2), Pearson's r (2), Euclidean distance (2), J-S alpha (32)	2, S.D. (1), 5, 9, Chi-stat (2), KS-stat (2), J-S alpha (2)	1, 3, Mean (1), Chi-stat (1), Pearson's r(1), K-L distance (1), 4, 5, 7
Low	Mean (1), Mean (2), Chi-stat(2), K-L distance (2), Cum_entropy_diff (2)	Chi-stat (1), Euclidean distance (1), J-S alpha (1), 0, 1, 4, 6, KS-stat (2), Euclidean distance (2)	2, 7, 9, Pearson's r (1), 0, 6 Chi-stat (2), KS-stat (2), J-S alpha (2)

and construct better-refined information theoretic and goodness-of-fit measures to be used as input variables in ANN-based systems.

As we have already stated, a binary logistic regression model may provide an alternative methodological approach distinct from neural networks if the problem of multicollinearity can be adequately addressed. Due to large-scale redundancies in their architecture, ANNs are better able to choose among explanatory variables in situations where multicollinearity is present as compared to alternative data classification systems like logistic regression and discriminant analysis [13].

Of course, ANNs are not an unmixed blessing as a data mining methodology. B&W [7] list some of the common drawbacks of using an ANN-based decision support tool most of which would be applicable to our ANN-based system as well. A potential drawback of this work lies in the use of simulated rather than actual financial data. Unfortunately it is extremely difficult to obtain enough quantity of real-life data that contains some degree of 'contamination'. There are privacy issues as corporations don't like to disseminate information on in-house frauds in fear of hurting their public image and possibly their market value. Therefore much of the research on development of effective data mining tools for ARP must necessarily rely on simulated data which, to some extent, brings their generalizability into question. However, we contend that there are other sciences (notably the atmospheric and geophysical sciences) where much of the research data must also be generated by simulation as real-life data may not simply be obtainable. We have followed the methodological footsteps of B&W in using simulated data and we cite a few well-regarded published works to back our chosen methodology [5,18,19].

With respect to the GA optimization, we acknowledge that the population size and number of generations is rather low. However we have used GA to primarily demonstrate its efficacy in setting up the optimal ANN structures without a compelling need to perform comparative analyses with alternative structures. GAs take a long time to converge to optimal solutions and this is further compounded by the highly complex and unstructured nature of the network optimization problem. Running the GAs with a larger population of networks or over a large number of generations will potentially offset the efficiency gains of using a GA in the first place to set up the optimal ANNs. However having said that we do believe there are ample areas of further improvement in the way that we have applied GA optimization to setup our ANNs and this provides fertile ground for future work. The relationship between insider trading and financial statements fraud has already been computationally investigated [40]. An interesting related future research from an applied perspective will be to test whether Benford's law in conjunction with a data mining system can be used to identify financial securities price anomalies due to dubious stock market activities like insider trading, pumping and dumping and market cornering.

Acknowledgements

The authors wish to acknowledge the valuable suggestions given by the Associate Editor as well as the anonymous reviewers in greatly improving the quality and readability of this work. Thanks are also due to Zoran Jasak of NLB Tuzlanska banka d.d., Tuzla and Renato M. Alas of University of Alaska, USA for their insightful comments that helped eliminate a few technical errors from the final draft.

Appendix 1. ANNs by Neuralyst™ v1.4

ANNs are a class of computational systems inspired by how the human brain works; primarily with respect to pattern learning and problem solving tasks. Although there is a whole range of possible neural network structures, the backpropagation network is by far the most widely applied in business and social science applications and is also the default one for Neuralyst™ v1.4 software [38]. A fully connected backpropagation multi-layer network is one where each neuron is connected to every output from the previous layer or the external domain if it is the input layer [38]. Running as a MS Excel add-in, Neuralyst™ v1.4 requires that the columns of an Excel spreadsheet individually represent different facts, goals or predictions for every problem instance [38]. The leading columns conventionally contain the data on the input variables (akin to the independent or "x" variables in a multiple regression model), followed by the 'target' column which contains data on the output variable (which is akin to the dependent or 'y' variable in a multiple regression model). This is conventionally followed by the output column which will hold the neural network output. This is conventionally followed by the 'mode flag' column which describes to the neural network whether a particular data item belongs to the training set or the test set. Only the data items that belong to the training set are used while training the neural network. Once the network has reached a certain pre-established training limit, the training effectiveness can be tested using the test set data items (provided the values of the output variable is known to the user for the test set data items). Once the input, target, output and mode flag columns have been specified, the size and configuration of the desired network can be specified to Neuralyst™, followed by setting the network parameters (like learning rate, momentum and input noise) and additional parameters (like the choice of the transfer function to be used to map the inputs to the output). Only then can training begin for the specified neural network. Interested readers are referred to Neuralyst version 1.4: User's Guide [38] for more details on the Neuralyst™ software.

Appendix 2. GA optimization by Neuralyst™ v1.4

Genetic algorithm (GA) optimization is a class of numerical optimization techniques that emulates the process of biological evolution via the mechanism of natural selection. This is especially suited for optimization problems that are ill-posed and unstructured. In a biological system, genetic material is encapsulated within *chromosomes*. This translates to *strings* for a genetic optimization algorithm like the one that is employed by the Neuralyst™ software. The entire genetic ensemble of chromosomes is called a *genotype*. In the Neuralyst™ genetic optimizer, there are three strings, one for input columns, one for network configuration and one for network parameters; the combination of all three representing a *structure*, which can define a neural network architecture [38]. The expression of a genotype as a distinct biological form is called a phenotype which corresponds to the expression of a structure as one of the many *candidate solutions*. The Neuralyst™ genetic optimizer evolves successive populations (generations) from a limited pool of starting candidate solutions. The input columns included, number of layers and number of neurons per layer and the values assigned for network parameters like learning rate and momentum are varied in each new generation with the resulting network evaluated in terms of its fitness with respect to some chosen fitness criterion. The default one for the Neuralyst™ software is the RMS training error [38]. Each structure within a generation is evaluated by either the lowest RMS error achieved after a set number of epochs or by the number of epochs taken to achieve a minimum RMS error level. If the structure successfully meets the fitness criterion specified "then the values of its feature will be retained and bred with other structures" [38]. All the input columns may not all be relevant for the learning task at hand or there may be redundancies in terms of their information content. The GA optimizer in Neuralyst™ v1.4 performs its evolutionary optimization by either including or excluding a particular input column. The inclusion rate for input data can vary between 1 and 100% while the default inclusion rate is 75%. Again, the interested reader is encouraged to look up Neuralyst version 1.4:

User's Guide [38] for a more complete understanding of the programmatic mechanics of the GA network optimizer that is built into Neuralyst™ v1.4.

References

[1] M.S. Beasley, An empirical analysis of the relation between the board of director composition and financial statement fraud, The Accounting Review 71 (4) (1996).
[2] F. Benford, The law of anomalous numbers, Proceedings of the American Philosophical Society 78 (4) (1938).
[3] S. Bhattacharya, K. Kumar, Forensic accounting and Benford's law, IEEE Signal Processing Magazine 25 (2) (2008).
[4] S. Bhattacharya, F. Smarandache, K. Kumar, Conditional probability of actually detecting a financial fraud—a neutrosophic extension to Benford's law, International Journal of Applied Mathematics 17 (1) (2005).
[5] R. Bieker, Using simulation as a tool in selecting a retirement age under defined benefit pension plans, Journal of Economics and Finance 26 (3) (2002).
[6] S.E. Bonner, Z.V. Palmrose, S.M. Yong, Fraud type and auditor litigation: an analysis of SEC accounting and auditing enforcement releases, The Accounting Review 73 (4) (1998).
[7] B. Busta, R. Weinberg, Using Benford's law and neural networks as a review procedure, Managerial Auditing Journal 13 (6) (1998).
[8] C. Carslaw, Anomalies in income numbers: evidence of goal oriented behaviour, The Accounting Review 63 (2) (1988).
[9] W.K.T. Cho, B.J. Gaines, Breaking the (Benford) law: statistical fraud detection in campaign finance, The American Statistician 61 (3) (2007).
[10] J.R. Coakley, C.E. Brown, Artificial neural networks applied to ratio analysis in the analytical review process, Intelligent Systems in Accounting, Finance and Management 2 (1993).
[11] J.R. Coakley, C.E. Brown, Artificial neural networks in accounting and finance: modeling issues, Intelligent Systems in Accounting, Finance and Management 9 (2000).
[12] C. Dalal, Numbers that do not add up, Business India, , Jan 2000.
[13] R.D. De Veaux, D.C. Psichogios, L.H. Ungar, A comparison of two non-parametric estimation schemes: MARS and neural networks, Computers in Chemical Engineering 17 (8) (1993).
[14] C. Durtschi, W. Hillison, C. Pacini, The effective use of Benford's law to assist in detecting fraud in accounting data, Journal of Forensic Accounting V (1) (2004).
[15] K. Fanning, K. Cogger, Neural network detection of management fraud using published financial data, Intelligent Systems in Accounting, Finance and Management 7 (1) (1998).
[16] K. Fanning, K. Cogger, R. Srivastava, Detection of management fraud: a neural network approach, Intelligent Systems in Accounting, Finance and Management 4 (2) (1995).
[17] E.H. Feroz, M.K. Taek, V.S. Pastena, K. Park, The efficacy of red flags in predicting the SEC's targets: an artificial neural networks approach, Intelligent Systems in Accounting, Finance and Management 9 (2000).
[18] E. Fishbein, C.B. Farmer, S.L. Granger, D.T. Gregorich, M.R. Gunson, S.E. Hannon, M. D. Hofstadter, S.-Y. Lee, S.S. Leroy, L.L. Strow, Formulation and validation of simulated data for the Atmospheric Infrared Sounder (AIRS), IEEE Transactions on Geoscience and Remote Sensing 41 (2) (2003).
[19] S. Gokhale, M.R. Lyu, K.S. Trivedi, Model validation using simulated data, Proceedings of Application-Specific Software Engineering and Technology (ASSET 98), Richardson, Texas, 1998.
[20] B.P. Green, J.H. Choi, Assessing the risk of management fraud through neural network technology, Auditing 16 (1997).
[21] T. Hill, Random-number guessing and the first digit phenomenon, Psychological Reports 62 (1988).
[22] S.-M. Huang, D.C. Yen, L.-W. Yang, J.-S. Hua, An investigation of Zipf's Law for fraud detection, Decision Support Systems DSS#06-10-1826R (2) (2008), doi:10.1016/j.dss.2008.05.003.
[23] G. Judge, L. Schechter, Detecting problems in survey data using Benford's law, Journal of Human Resources 44 (1) (2009).
[24] K.-J. Kim, S.-B. Cho, in: M. Ishikawa, et al., (Eds.), Diverse Evolutionary Neural Networks Based on Information Theory, ICONIP 2007 Part II, , 2007.
[25] E. Kirkos, C. Spathis, Y. Manolopoulos, Data mining techniques for the detection of fraudulent financial statements, Expert Systems with Applications 32 (4) (2007).
[26] S. Kullback, R.A. Leibler, On information and sufficiency, Annals of Mathematical Statistics 22 (1951).
[27] K. Kumar, S. Bhattacharya, Benford's law and its application in financial fraud detection, in: C.F. Lee (Ed.), Advances in Financial Planning and Forecasting, 11, 2003.
[28] K. Kumar, S. Bhattacharya, Detecting the dubious digits: Benford's law in forensic accounting, Significance: Magazine of the Royal Statistical Society 4 (2) (2007).
[29] T.A. Lee, R.W. Ingram, T.P. Howard, The difference between earnings and operating cash flow as an indicator of financial reporting fraud, Contemporary Accounting Research 16 (4) (1999).
[30] J.W. Lin, M.I. Hwang, J.D. Becker, A fuzzy neural network for assessing the risk of fraudulent financial reporting, Managerial Auditing Journal 18 (2003).
[31] M.J. Nigrini, A taxpayer compliance analysis of Benford's law, Journal of the American Taxation Association 18 (1) (1996).
[32] M.J. Nigrini, Digital Analysis Tests and Statistics, The Nigrini Institute Inc., Allen, Texas, 1997.
[33] M.J. Nigrini, I've got your number, Journal of Accountancy 187 (1999).
[34] A.N. Petitt, M.A. Stevens, The Kolmogorov–Smirnov goodness-of-fit statistics for discrete and grouped data, Technometrics 19 (1977).
[35] L. Pietronero, E. Tosatti, V. Tosatti, A. Vespignani, Explaining the uneven distribution of numbers in nature: the laws of Benford and Zipf, Physica A 293 (2004).
[36] M. Ramos, Auditors' responsibility for fraud detection: SAS no. 99 introduces a new era in auditors' requirements, Journal of Accountancy Online (2003) www.journalofaccountancy.com/Issues/2003/Jan/, last retrieved on 15/12/2008.
[37] C.E. Shannon, A mathematical theory of communication, Bell System Technical Journal 27 (1948).
[38] Y. Shih, Neuralyst version 1.4: user's guide, Cheshire Engineering Corp., Pasadena, 1994.
[39] M. Steele, J. Chaseling, Powers of discrete goodness-of-fit test statistics for a uniform null against a selection of alternative distributions, communications in statistics, Simulation and Computation 35 (2006).
[40] S.L. Summers, J.T. Sweeney, Fraudulent misstated financial statements and insider trading: an empirical analysis, The Accounting Review 73 (1) (1998).
[41] J. Thomas, Unusual patterns in reported earnings, The Accounting Review 64 (4) (1989).
[42] C. Watrin, R. Struffert, R. Ullmann, Benford's law: an instrument for selecting tax audit targets, Review of Managerial Science 2 (3) (2008).
[43] O.J. Welch, T.E. Reeves, S.T. Welch, Using a genetic algorithm based classifier system for modeling auditor decision behavior in a fraud setting, Intelligent Systems in Accounting, Finance and Management 7 (3) (1998).
[44] A. Wright, R.H. Ashton, Identifying audit adjustments with attention-direction procedures, The Accounting Review 64 (4) (1989).
[45] P.E. Zikmund, Reducing the expectation gap: forensic audit procedures, The CPA Journal Online (2008) www.nysscpa.org/cpajournal/2008/608/essentials/p20.htm, last retrieved on 15/12/2008.
[46] G.K. Zipf, Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology, Hafner Publications, NY, 1949.

Dr. Sukanto Bhattacharya received his PhD from the School of Information Technology, Bond University, Australia in 2004. He has served as an Assistant Professor of Finance at Alaska Pacific University, Anchorage, USA and also at Dickinson College, Pennsylvania, USA prior to returning to Australia in 2008 and subsequently joining as a Postdoctoral Research Fellow at University of Queensland's UQ-KPMG Centre for Business Forensics. He is currently a Senior Lecturer in Finance at Deakin Business School, Deakin University, Australia. Sukanto won the faculty excellence award in 2005–2006 for outstanding contribution to research, scholarship and creativity at Alaska Pacific University, USA. He also features among the Who's Who in Collegiate Faculty. His main area of research interest concerns mathematical models for financial fraud detection, business failure prediction and forecasting/analysis of corporate credit ratings.

Dr. Dongming Xu is a Senior Lecturer in Information Systems at Business School, The University of Queensland. She holds a PhD from the City University of Hong Kong in Information systems. Xu's main research areas are business intelligence and service computing, investigating topics such as decision support systems for securities exception management, intelligent business activities monitoring, web service agents-based Family Wealth Management Systems, knowledge-based intelligent money laundering monitoring systems, etc. Her research on these areas is focused on a variety of contexts, for example theoretical foundations, conceptual and ontological modelling, applications, and technologies such as intelligent agents and data mining.

Dr. Kuldeep Kumar is currently Professor and Head, Department of Economics and Statistics at Bond University, Gold Coast, Australia. He did his PhD in the University of Kent, UK and has taught at Indian Institute of Management, Lucknow and the National University of Singapore before joining Bond University. Dr. Kumar is a winner of several awards including Commonwealth Scholarship, Bond-Oxford Fellowship, Young Statistician Award and VC quality award, Dr Kumar has published around one hundred papers. His current research interest is in the area of Business Intelligence including bankruptcy prediction, financial fraud detection and economic crime prevention applying statistical, computational and data mining methods and tools.