Volume 2 No.4, APRIL 2011 ISSN 2079-8407
Journal of Emerging Trends in Computing and Information Sciences

http://www.cisjournal.org

Detecting Auto Insurance Fraud by Data Mining Techniques

Rekha Bhowmik
Computer Science Department
University of Texas at Dallas, USA
rxb080100@utdallas.edu

ABSTRACT

The paper presents fraud detection method to predict and analyze fraud patterns from data. To generate classifiers, we apply the Naïve Bayesian Classification, and Decision Tree-Based algorithms. A brief description of the algorithm is provided along with its application in detecting fraud. The same data is used for both the techniques. We analyze and interpret the classifier predictions. The model prediction is supported by Bayesian Naïve Visualization, Decision Tree visualization, and Rule-Based Classification. We evaluate techniques to solve fraud detection in automobile insurance.

Keywords: *Rule-based Algorithm, Bayesian Network, C4.5, Fraud Detection*

1. INTRODUCTION

There are a number of data mining techniques like clustering, neural networks, regression, multiple predictive models. Here, we discuss only few techniques of data mining which would be considered important to handle fraud detection.

Data Mining is associated with (a) supervised learning based on training data of known fraud and legal cases and (b) *unsupervised learning* with data that are not labeled to be fraud or legal. Bedford's law can be interpreted as an example of unsupervised learning [1].

Insurance fraud, credit card fraud, telecommunications fraud, and check forgery are some of the main types of fraud. Insurance fraud is common in automobile, travel. Fraud detection involves three types of offenders: i) Criminal offenders, ii) organized criminal offenders who are responsible for major fraud, and iii) offenders who commit fraud (called soft fraud) when suffering from financial hardship. Soft fraud is the hardest to lessen because the cost for each suspected incident is usually higher than the cost of the fraud. Types i) and ii) offenders, called hard fraud, avoid anti-fraud measures [2].

We present data mining techniques which are most appropriate for fraud analysis. We present automobile insurance example. Here, the data mining techniques used for fraud analysis are: i) Bayesian network, and ii) Decision tree. Bayesian network is the technique used for classification task. Classification, given a set of predefined categorical classes, determines which of these classes a specific data belongs to. Decision trees are used to create descriptive models. Descriptive models are created to describe the characteristics of fault.

The remainder of this paper is organized as follows. In Section 2, we present the existing fraud detection systems and techniques. Section 3 includes the algorithms and application. Section 4 presents the model. Finally, in section 5, we discuss the important features of our work.

2. EXISTING FRAUD DETECTION SYSTEMS

The hot spots methodology[3] performed a three step process: i) k-means clustering algorithm for cluster detection is used because the other clustering algorithms tend to be expensive for very large datasets, ii) C4.5 algorithm, the resulting decision tree can be converted to a rule set and pruned, and iii) visualization tools for rule evaluation, building statistical summaries of the entities associated with each rule.

The credit fraud model[4] suggested a classification technique with fraud/legal attribute, and a clustering followed by a classification technique with no fraud/legal attribute. Kohonen's Self-Organizing Feature Map [5] was used to categorize automobile injury claims depending on the type of fraud.

Classification techniques have proved to be very effective in fraud detection[6] and therefore, can be applied to categorize crime data. The distributed data mining model[6] uses a realistic cost model to evaluate C4.5, CART, and naïve Bayesian classification models. The method was applied to credit card transactions. The neural data mining approach[7] uses rule-based association rules to mine symbolic data. The approach discusses the importance of use of non-numeric data in fraud detection.

SAS Enterprise Miner Software[8] depends on association rules, cluster detection and classification techniques to detect fraudulent claims. The Bayesian Belief Network (BBN) and Artificial Neural Network (ANN) study used the STAGE algorithm for BBN in fraud detection and backpropagation for ANN[9]. The result shows that BBNs were much faster to train, but were slower when applied to new instances. The ASPECT group[10] focused on neural networks to train current user profiles and user profiles histories. A caller's current profile and the profile history are compared to find probable fraud. [11] build on the adaptive fraud detection framework[12, 13], by applying an event-driven approach of assigning fraud scores to detect fraud. The [11] framework can also detect types of fraud using rules. [14]

used dynamic BBNs called Mass Detection tool to detect fraudulent claims, which then used a rule generator called Suspicion Building Tool.

Internal fraud detection consists in determining fraudulent financial reporting by management[15], and abnormal retail transactions by employees[16]. There are four types of insurance fraud detection: home insurance[17], crop insurance [18], automobile insurance fraud detection[19], and health insurance[20]. A single meta-classifier[21] is used to select the best base classifiers, and then combined with these base classifiers' predictions to improve cost savings. Credit card fraud detection refers to screening credit applications, and/or logged credit card transactions [22]. Credit transactional fraud detection has been presented by [22]. Literature focus on video-on-demand websites[23] and IP-based telecommunication services[24]. Online sellers[25] and online buyers[26] can be monitored by automated systems. Fraud detection in government organisations such as tax[27] and customs[28] has also been reported.

2.1 Bayesian Belief Networks

Naïve Bayesian classification assumes that the attributes of an instance are independent, given the target attribute[29]. The aim is to assign a new instance to the class that has the highest posterior probability. The algorithm is very effective and can give better predictive accuracy when compared to C4.5 decision trees and backpropagation

2.2 Decision Trees

Decision trees are machine learning techniques that express independent attributes and a dependent attribute in a tree-shaped structure. Classification rules, extracted from decision trees, are IF-THEN expressions in which the preconditions are logically ANDed and all the tests have to succeed if each rule is to be generated. The related applications include the analysis of instances from drug smuggling, governmental financial transactions[30], and customs declaration fraud[28] to more serious crimes such as drug related homicides, serial sex crimes[31], and homeland security[31, 30]. C4.5 [32] is used to divide data into segments based and to generate descriptive classification rules that can be used to classify a new instance. C4.5 can help to make predictions and to extract crime patterns. It generates rules from trees [33] and handles numeric attributes, missing values, pruning, and estimating error rates. The learning and classification steps are generally fast. However, performance decrease can occur when C4.5 is applied to large datasets. C5.0 shows marginal improvements to decision tree induction.

3. APPLICATION

The steps in crime detection are: i) classifiers, ii) integrate multiple classifiers, iii) ANN approach to clustering, and iv) visualization techniques to describe the patterns.

3.1 Bayesian Network

For the purpose of fraud detection, we construct two Bayesian networks to describe the behavior of auto insurance. First, a Bayesian network is constructed to model behavior under the assumption that the driver is fraudulent and another model under the assumption the driver is a legal. The fraud net is set up by using expert knowledge. The legal net is set up by using data from legal drivers. By inserting evidence in these networks, we can get the probability of the measurement E under two above mentioned hypotheses. This means, we obtain judgments to what degree an observed user behavior meets typical fraudulent or legal behavior. These quantities we call P(E| output = legal) and P(E| output = fraud). By postulating the probability of fraud P(output = fraud) and P(output = legal) = 1 - P(output = fraud) in general and by applying Bayes' rule, we get the probability of fraud, given the measurement E,

P(output = fraud | E) = P(output = fraud) P(E| output = fraud) / P(E)

where, the denominator $P(E)$ can be calculated as:

$P(E) = P$(output = fraud) P(E| output = fraud) + P(output = legal) P(E| output = legal)

The chain rule of probabilities is:

Suppose there are two outputs O_1, O_2 for *fraud* and *legal* respectively. Given an instance $E = (E_1, E_2, \ldots, E_n)$, each row is represented by an attribute $A = (A_1, A_2, \ldots, A_n)$

The classification is to derive the maximum $P(O_i|X)$ which can be derived from Bayes' theorem.

3.1.1 Application

We present Bayesian learning algorithm to predict occurrence of fraud.

Consider the two output attributes, *fraud* and *legal*. The general equation for computing the probability that the output attribute is *legal* or *fraud* is:

i) P(output = fraud | E)
= [P(E | output = fraud) P(output = fraud)] / P(E
P(output = legal | E) = [P(E | output = legal) P(output = legal)] / P(E)

ii) The a priori probability, shown as P(output=fraud), is the probability of a *fraud* customer without knowing the history of the instance. Here, the a priori probability is the fraction of the total population that is *fraud*, that is:
P(fraud) = d_i / d
d is the total population and d_i is the number of *fraud.*

iii) A simplified assumption of no dependent relationships between attributes is made.

Thus,

$$P(E|\text{ output = fraud}) = \prod_{k=1}^{n} P(x_k|\text{output = fraud})$$

$$P(E|\text{ output = legal}) = \prod_{k=1}^{n} P(x_k|\text{output = legal})$$

The probabilities $P(x_1|\text{output} = \textit{fraud})$, $P(x_2|\text{output} = \textit{fraud})$ can be estimated from the database using:

$P(x_k|\text{output = fraud}) = d_{ik} / d_i$

Here, d_i is the number of records for output *fraud* and d_{ik} is the number of records of output class *fraud* having the value x_k for the attributes.

iv) Repeat step iii) for computing

$P(E \mid \text{output = legal})$

$[P(E \mid \text{output = fraud})\ P(\text{output = fraud})]$ and

$[P(E \mid \text{output = legal})\ P(\text{output = legal})]$ need to be optimized as $P(E)$ is constant .

Consider the data in Table 1, which is a subset of auto insurance database. We use "Output" attribute whose value is to be predicted.

E= (policyHolder = 1, driverRating = 0, reportFiled = 0.33) to be either *fraud* or legal.

$P(\text{fraud}) = d_i / d = 3/20 = 0.15$

$P(\text{legal}) = d_i / d = 17/20 = 0.85$

From step iii) of the algorithm,

$P(\text{policyHolder = 1/ output=fraud}) = 3/3 = 1$

$$P(E|\text{ output = fraud}) = \prod_{k=1}^{n} P(x_k|\text{output = fraud}) = 0$$

From step iv) of the algorithm,

$P(\text{policyholder = 1/ output=legal}) = 12/17 = 0.706$

$$P(E|\text{ output = fraud}) = \prod_{k=1}^{n} P(x_k|\text{output = legal})$$

$= 0.0068$

Therefore,

$[P(E \mid \text{output = fraud})\ P(\text{output = fraud})] = 0$

$[P(E \mid \text{output = legal})\ P(\text{output = legal})] = 0.0058$

Based on these probabilities, we classify the new tuple as *legal*. The probabilities for $P(E|\text{output = fraud})$ is always 0. The Laplace estimator improves the value by adding 1 to the numerator and the total number of attribute value types to the denominator of $P(E|\text{output = fraud})$ and $P(E|\text{output = fraud})$ [33].

Based on step iii) of the algorithm,

$P(\text{policyHolder = 1/ output=fraud}) = 0.8$

From step iv) of the algorithm,

$P(\text{policyholder = 1/ output=legal}) = 0.684$

$[P(E \mid \text{output = fraud})\ P(\text{output = fraud})] = 0.0026$

$[P(E \mid \text{output = legal})\ P(\text{output = legal})] = 0.0016$

Thus, instance E is more likely to be *Fraud*.

Likelihood of being *legal* =0.0351

Likelihood of being *fraud* = 0.050

We estimate $P(E)$ by summing up these individuals likelihood values since E will be either *legal* of *fraud*:

$P(E) = 0.0351 + 0.050 = 0.0851$

Finally, we obtain the actual probabilities of each event:

$P(\text{output = legal} \mid E) = (0.039 * 0.9)/ 0.0851 = 0.412$

$P(\text{output = fraud} \mid E) = (0.500 * 0.1)/ 0.0851 = 0.588$

Bayesian classifier can handle missing values in training datasets. To demonstrate this, seven missing values appear in dataset.

The Naïve Bayes approach is easy to use and only one scan of the data is required. The approach can handle missing values by simply omitting that probability when calculating the likelihoods of membership in each class.

3.2 Decision Tree-Based Algorithm

Solving the classification problem is a two-step process: i) decision tree induction- construct a Decision Tree(DT), and ii) apply the DT to determine its class. Rules can be generated that are easy to interpret.

The basic algorithm for decision tree is as follows:

i) Suppose there are two outputs, *fraud* and *legal*. The tree starts as a single node N representing the dataset. If the instances are of the same type *fraud*, then the node becomes a leaf and is labeled as *fraud*.

ii) Otherwise, the algorithm uses an Entropy, Gini Index, and Classification Error to measure degree of impurity for selecting the attribute that will best separate the data into individual classes.

iii) Entropy is calculated as the sum of the conditional probabilities of an event (p_i) times its information required for the event in subsets (b_i). Note that $b_i = -\log_2 p_i$ in the cases of a simple (binary) split into two classes.

$$\text{Entropy}(p_1,p_2,\ldots,p_n) = p_1 * b_1 + p_2 * b_2 \ldots + p_n * b_n$$
$$= -p_1\log p_1 - p_2\log p_2 \ldots - p_n\log p_n$$

Table 1a. Data for Bayes Classifier

instance	Policy Holder	Driver Rating	Report Filed	Output
1	1	0	0	legal
2	1	1	1	fraud
3	0	0	0	legal
4	1	0.33	1	legal
5	1	0.66	0	legal
E	1	0	0.33	?

Table 1b. Data for Bayes Classifier

instance	Policy Holder	Driver Rating	Report Filed	Vehicle AgePrice	Output
1	1	0	0	0.33	legal
2	1	1	1	0.5	fraud
3	0	0	0	0.75	legal
4	1	0.33	1	0.5	legal
5	1	0.66	0	0.5	legal
E	1	0	0.33	0.5	?

3.2.1 C4.5 Algorithm

The Entropy, or expected information needed to classify a given instance is:

P(fraud, legal)= – (fraudInstances / Instances) $\log_2$ (fraudInstances / Instances) – (legalInstances / Instances) $\log_2$ (legalInstances / Instances)

Expected information or entropy by attribute:

E(*A*)= ∑ [{(fraudAttributes / Instances) + (legalAttributes/ Instances)} * {*E*(fraudAttributes, legalAttributes)}]

iv) The value (or contribution to information) of an attribute is calculated as *gain*(attr) = (information before split) - (information after split)

Expected reduction in entropy is:

gain(attr) = *Entropy of parent table*– *E*(*A*)

The algorithm computes the information gain of each attribute. The attribute with the highest information gain is the one selected for test attribute.

v) A branch is created for each known value of the test attribute. The algorithm uses the same process iteratively to form a decision tree at each partition. Once an attribute has occurred at a node, it need not be considered in any of the node's descendents.

vi) The iterative partitioning stops when one of the conditions is true: a) all examples for a given node belong to the same class, or b) there are no remaining attributes on which samples may be further partitioned, and c) there are no samples for the branch test-attribute.

3.2.2 Application

From Table 1b, the probability of each output class is:

entropy = -0.1 log (0.1) – 0.9log(0.9) = - 0.1*3.32 -0.9* 0.152 =0 .469

E(vehicleAgePrice) = (9/20) entropy(1, 8) = (9/20) (-1/9 $\log_2$ 1/9 - 8/9 $\log_2$ 8/9) = .225

The information gain of attribute VehicleAgePrice is computed as follows:

0.469 – [(9/20) (-1/9 $\log_2$ 1/9 - 8/9 $\log_2$ 8/9)] = 0.244

prob(output = fraud) = 2/20 = 0.1

gini index $= 1 - \sum_j (\text{prob})_j^2$

$= (0.1^2 + 0.9^2) = 0.18$

Classification error = 1- max{prob $_j$ }

= 1- max{0.1, 0.9} = 0.9

Entropy, Gini Index, and Classification Error Index of single class is zero. They reach maximum value when all the classes in the table have equal probability.

The attribute VehicleAgePrice has four values. Based on step v) of C4.5 algorithm, a decision tree can be created. Each node is either i) a leaf node - (output class), or ii)a decision node

3.3 Rule Based Algorithm

One way to perform classification is to generate if-then rules.

3.3.1 Generating Rules from a Decision Tree

The following rules are generated for the Decision Tree:

If (driver_age ≤ 40)) ∧ (driver_rating =1)) ∧ (vehicle_age =2), then class = fraud

If (driver_age > 40)) ∧ (driver_age ≤ 50)) ∧ (driver_rating = 0.33), then class = legal

4. MODEL PERFORMANCE

Confusion Matrix

There are two ways to examine the performance of classifiers: i) confusion matrix, and ii) to use a ROC graph. Given a class, C_j, and a tuple, t_i, that tuple may or may not be assigned to that class while its actual membership may or may not be in that class. With two classes, there are four possible outcomes with the classification as: i) true positives (hits), ii) false positives (false alarms), iii) true negatives (correct rejections), and iv) false negatives. Table 2a, contains information about actual and predicted classifications. Performance is evaluated using the data in the matrix. Table 2b shows confusion matrix built on simulated data. The model commits some errors and has an accuracy of 78%. We also applied the model to the same data, but to the negative class with respect to class skew in the data. The quality of a model highly depends on the choice of the test data. A number of model performance metrics can be derived from the confusion matrix.

Table 2a. Confusion Matrix

		Observed	
		legal	fraud
predicted	legal	TP	FP
	fraud	FN	TN

Table 2b. Confusion matrix of a model applied to test dataset

		Observed		
		legal	fraud	accuracy: 0.78
predicted	legal	3100	1125	recall: 0.86
	fraud	395	2380	precision: 0.70

The accuracy determined in (Table 2b) may not be an adequate performance measure when the number of negative cases is much greater than the number of positive cases. Suppose there are 1500 cases, 1460 of which are negative cases and 40 of which are positive cases. If the system classifies them all as negative, the accuracy would be 97.3%, even though the classifier missed all positive cases. Other performance measures are geometric mean (g-mean), and F-Measure. For calculating F-measure, β has a value from 0 to ∞ and is used to control the weight assigned to TP and P. Any classifier evaluated using g-mean or F-measure will have a value of 0, if all positive cases are classified incorrectly.

To easily view and understand the output, visualization of the results is helpful.

Naïve Bayesian visualization provides an interactive view of the prediction results. The attributes can be sorted by the predictor and evidence items can be sorted by the number of items in its storage bin. Attribute column graphs help to find the significant attributes in neural networks. Decision tree visualization builds trees by splitting attributes from C4.5 classifiers.

Cumulative gains and lift charts are visual aids for measuring model performance. Lift is a measure of a predictive model calculated as the ratio between the results obtained with or without the predictive model. For instance, if 105 of all samples are actually fraud and a naïve Bayesian classifier could correctly predict 20 fraud samples per 100 samples, then that corresponds to a lift of 4.

Table 3c: Performance metrics

model performance metrics	
Accuracy(AC)	$AC = \frac{a+d}{a+b+c+d}$
Recall or true positive rate(TP)	$TP = \frac{d}{c+d}$
False positive rate(FP)	$FP = \frac{b}{a+b}$
True negative rate(TN)	$TN = \frac{a}{a+b}$
False negative rate(FN)	$FN = \frac{c}{c+d}$
Precision(P)	$P = \frac{d}{b+d}$
geometric mean(g-mean)	$g-mean_1 = \sqrt{TP*P}$ $g-mean_2 = \sqrt{TP*TN}$
F-measure	$F = \frac{(\beta^2+1)*P*TP}{\beta^2*P+TP}$

Classification models are often evaluated on accuracy rates, error rates, false negative rates, and false positive rates.

Table 3 shows that True Positives (hits) and False Positives (false alarms) require cost per investigation. False alarms cost are the most expensive because both investigation and claim costs are required. False Negatives (misses) and True Negatives(correct rejection) are the cost of claim.

Table 3: Cost/ Benefit Decision Summary of Predictions

fraud	legal
True Positive(Hit) cost = number of hits * average cost per investigation	False Positive(False alarm) cost =number of false alarms * (Average cost per investigation + average cost per claim)
False Negative(miss) cost = number of misses * average cost per claim	True Negative(correct rejection) cost = number of correct rejection claims * average cost per claim

5. CONCLUSIONS

We studied the existing fraud detection systems. To predict and present fraud we used Naïve Bayesian classifier and Decision Tree-Based algorithms. We looked at model performance metrics derived from the confusion matrix. Performance metrics such as accuracy, recall, and precision are derived from the confusion matrix. It is strong with respect to class skew, making it a reliable performance metric in many important fraud detection application areas.

REFERENCES

[1] Bolton, R., Hand, D.: Statistical Fraud Detection: A Review. Statistical Science 17(3): 235--255(2002).

[2] Sparrow, M. K.: Fraud Control in the Health Care Industry: Assessing the State of the Art, in Shichor et al(eds), Readings in white-Collar Crime, Waveland Press, Illinois(2002).

[3] Williams, G.: Evolutionary Hot Spots Data Mining: An Architecture for Exploring for Interesting Discoveries. In: 3rd Pacific-Asia Conference in Knowledge Discovery and Data Mining, Beijing, China(1999).

[4] Groth, R.: Data Mining: A Hands-on Approach for Business Professionals, Prentice Hall, pp. 209-212(1998).

[5] Brockett, P., Derrig, R., Golden, L., Levine, A. & Alpert, M.: Fraud Classification using Principal Component Analysis of RIDITs. *Journal of Risk and Insurance* 69(3): 341-371(2002).

[6] Chen, R., Chiu, M., Huang, Y., Chen, L.: Detecting Credit Card Fraud by Using Questionnaire-Responded Transaction Model Based on Support Vector Machines. In: IDEAL2004, 800--806(2004).

[7] Brause, R., Langsdorf, T. , Hepp, M.: Neural Data Mining for Credit Card Fraud Detection. In: 11th IEEE International Conference on Tools with Artificial Intelligence(1999).

[8] SAS, e-IntelligenceData Mining in the Insurance industry: Solving Business problems using SAS Enterprise Miner Software. White Paper(2000).

[9] Maes, S., Tuyls, K., Vanschoenwinkel, B. & Manderick, B.: Credit Card Fraud Detection using Bayesian and Neural Networks. *Proc. of the 1st International NAISO Congress on Neuro Fuzzy Technologies* (2002).

[10] Weatherford, M.: Mining for Fraud. In: IEEE Intelligent Systems(2002).

[11] Cahill, M., Chen, F., Lambert, D., Pinheiro, J. & Sun, D.: Detecting Fraud in the Real World. *Handbook of Massive Datasets* 911-930(2002)

[12] Fawcett, T.: ROC graphs: Notes and practical considerations for researchers. Machine Learning, 3(2004).

[13] Fawcett, T., Flach, P. A.: A response to web and Ting's on the application of ROC analysis to predict classification performance under varying class distributions. Machine Learning, 58(1), 33--38(2005).

[14] Ormerod T., Morley N., Ball L., Langley C., Spenser C.: Using Ethnography To Design a Mass Detection Tool (MDT) for the Early Discovery of Insurance Fraud. Computer Human Interaction, Ft. Lauderdale, Florida(2003).

[15] Lin, J., Hwang, M. , Becker, J.: A Fuzzy Neural Network for Assessing the Risk of Fraudulent Financial Reporting. J. of Managerial Auditing, 18(8), 657--665(2003).

[16] Kim, H., Pang, S., Je, H., Kim, D. & Bang, S.: Constructing Support Vector Machine Ensemble. *Pattern Recognition* **36**: 2757-2767(2003).Kim, J., Ong, A. & Overill, R. (2003). Design of an Artificial Immune System as a Novel Anomaly Detector for Combating Financial Fraud in Retail Sector. *Congress on Evolutionary Computation*.

[17] Bentley, P., Kim, J., Jung., G., Choi, J.: Fuzzy Darwinian Detection of Credit Card Fraud. In: 14th Annual Fall Symposium of the Korean Information Processing Society(2000).

[18] Little, B., Johnston, W., Lovell, A., Rejesus, R. & Steed, S.: Collusion in the US Crop Insurance Program: Applied Data Mining. *Proc. of SIGKDD02*, 594-598(2002).

[19] Viaene, S., Derrig, R., Dedene, G.: A Case Study of Applying Boosting Naive Bayes to Claim Fraud Diagnosis. In: IEEE Transactions on Knowledge and Data Engineering, 16(5), 612--620(2004).

[20] Yamanishi, K., Takeuchi, J., Williams, G. , Milne, P.: On-Line Unsupervised Outlier Detection Using Finite Mixtures with Discounting Learning Algorithms. Data Mining and Knowledge Discovery, 8, 275--300(2004).

[21] Phua, C., Alahakoon, D., Lee, V.: Minority Report in Fraud Detection: Classification of Skewed Data. In: SIGKDD Explorations, 6(1), 50--59(2004).

[22] Foster, D. & Stine, R.: Variable Selection in Data Mining: Building a Predictive Model for Bankruptcy. *J. of American Statistical Association* 99, 303--313(2004).

[23] Barse, E., Kvarnstrom, H. , Jonsson, E.: Synthesizing Test Data for Fraud Detection Systems. In: 19th Annual Computer Security Applications Conference, 384--395(2003).

[24] McGibney, J., Hearne, S.: An Approach to Rules-based Fraud Management in Emerging Converged Networks. In: IEI/IEEE ITSRS (2003).

[25] Bhargava, B., Zhong, Y., Lu, Y.: Fraud Formalization and Detection. In: *DaWaK2003*, 330--339(2003).

[26] Sherman, E.: Fighting Web Fraud. Newsweek, June 10(2002).

[27] Bonchi, F., Giannotti, F., Mainetto, G., Pedreschi, D.: A Classification-based Methodology for Planning Auditing Strategies in Fraud Detection. In: SIGKDD99, 175--184(1999).

[28] Shao, H., Zhao, H., Chang, G.: Applying Data Mining to Detect Fraud Behavior in Customs Declaration. In: 1st International Conference on Machine Learning and Cybernetics, 1241--1244(2002).

[29] Feelders, A. J.: Statistical Concepts. Berthold M. and Hand D. (eds), Intelligent Data Analysis, Springer-Verlag, Berlin, Germany, pp. 17-68, 2003.

[30] Mena J.: Data mining for Homeland Security. Executive Briefing, VA(2003).
Mena J.: Investigative Data Mining for Security and Criminal Detection, Butterworth Heinemann, MA(2003).

[31] SPSS: Data mining and Crime analysis in the Richmond Police Department, White Paper, Virginia(2003).

[31] James F.: FBI has eye on business databases. *Chicago Tribune*, Knight Ridder/ Tribune Business News(2002).

[32] Quinlan, J. R.: C4.5 Programs for Machine Learning, Morgan Kauffman, CA, USA(1993).

[33] Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd Edition, Morgan Kaufmann(2005).