Edwards

Journal of Forensic Accounting
1524-5586 / Vol.VIII (2007), pp.

Correlation of Complex Evidence in Forensic Accounting Using Data Mining

Boris Kovalerchuk,[1*] **Evgenii Vityaev**[2] and **Robert Holtfreter**[1]

[1]Central Washington University, Ellensburg, WA USA
[2]Russian Academy of Science, Novosibirsk Russia

The classical statistical correlation is an efficient technique for linking simple numerical data sets via a single correlation coefficient. The modern schemes for money laundering and financial fraud are becoming very sophisticated and are changed all the time. To be able to discover such schemes we need to deal simultaneously with a diverse set of numeric and non-numeric data types that include different numeric data types, ordered sets, graph structures, texts, schemes, plans, and other information. Often any individual evidence does not reveal a suspicious pattern and does not guide investigation in forensic accounting. In contrast, correlation of two or more evidences with each other and background knowledge can reveal a suspicious pattern. The area of Link Discovery (LD) has emerged recently as a promising new approach for such tasks. This paper outlines design of such a new technique called Hybrid Evidence Correlation (HEC). It combines first-order logic (FOL), probabilistic semantic inference (PSI) and negative rules for designing HEC to deal with rare suspicious patterns. The approach is illustrated with an example of discovery of suspicious patterns. Computational efficiency of the algorithm is justified by a computational experiment. Conceptual advantages of the algorithm such as completeness have been reported in previous mathematical analysis of the base concepts of the algorithm. The approach was successfully tested for detecting fraudulent transactions in synthetic data. Data contained several attributes of a transaction such as seller, buyer, types of buyer and seller, sold item, amount, price and date.

INTRODUCTION

Prior to recent developments in technology, managers, tips from employees and internal auditors detected most occupational frauds. Another current focus in forensic accounting is the analysis of funding mechanisms for terrorism [Prentice, 2002] where *clean money* (e.g., charity money) *and laundered money are both used* for a variety of activities including acquisition and production of weapons and their precursors. This is in contrast with traditional illegal businesses and drug trafficking that *make dirty money appear clean*.

The area of Link Discovery (LD) has emerged recently as a promising new approach for such tasks. Potential applications of link discovery range from basic science to a variety of practical forensic tasks. Currently LD mostly relies on deterministic graphical techniques. Bayesian probabilistic and causal networks are other relevant techniques. Both techniques need further development to handle rare events. Complexity of the task dictates the need to combine the statistical techniques with other mathematical techniques.

Before a possible fraud can be investigated it must be detected. The process of fraud detection involves searching for symptoms that may indicate that fraud exists. To search for symptoms of fraud in databases, data mining methods can be very useful. The purpose of this article is to introduce an inductive data-mining method that may be used to search for suspicious patterns or anomalies in data that may represent symptoms leading to the detection of fraud. The specific tasks in automated forensic accounting are the identification of suspicious and unusual electronic transactions and the reduction in the number of 'false positive' suspicious transactions.

There are several challenges in automated transaction monitoring systems [Bolton, Hand, 2002; Rosenthal, 2001; Weatherford, 2002; FSO, 2002]: (1) building inexpensive, simple rules based systems and customer profiling; (2) reducing the number of 'false positive' suspicious transactions, and (3) fusing data from multiple sources to get a larger picture.

State-of-the-art

Currently inexpensive, simple rule-based systems, customer profiling, statistical techniques, neural networks, fuzzy logic, and genetic algorithms are considered as appropriate tools [Chartier, Spillane, 2000; Prentice, 2002]. Forensic accountants, attorneys and fraud examiners can use tools such as ACL, NetMap, Analyst's Notebook and others [Chabrow, 2002; i2, 2003; Evett, Jackson, Lambert, McCrossan, 2000].

For the last two decades ACL (Audit Command Language) [Will, 1983] and ACL based software have been used for auditing purposes to search for anomalies or suspicious patterns in databases [http://www.acl.com]. Using ACL auditors analyze payroll, employee expense accounts, accounts payable, and accounts receivable and others. For instance, the standard payment test runs a comparison between the total expected payments and the actual payments received. Exceptions are exported to a report for further investigation. Technically it is done by using the ACL expression builder tool that allows building a program with the terms of clients' contracts [http://www.findarticles.com/p/articles /mi_m4153 /is_3_58 /ai_77151364].

Another application of ACL can be illustrated by using a vendor fraud case where a member of a purchasing department (who has responsibility for ordering goods from vendors) is suspected of taking kickbacks from a vendor. The ACL can sort database records by vendor and volume to determine if total purchases from one vendor were increasing while overall purchases were stable or decreasing. Finding such a pattern could be a symptom of fraud and a fraud investigation could be initiated to determine if the suspected buyer was accepting kickbacks from the vendor.

The most important characteristics of ACL based systems are the need for (1) *programming of requirements* (e.g., based on contracts) and (2) testing databases against these requirements. This is useful for fraud detection, but it is limited by situations with clearly stated (written) requirements such as expected payments. These written requirements can be met, but fraud can be in place in violation of other less clearly identified requirements.

ACL based software is not data-mining software within the narrow definition that assumes that audit rules (requirements) are *learned from data*. In current ACL based software systems, requirements (rules) are *programmed by humans* and are not learned from data by a software system.

The major advantage of using a commercial software package like ACL is that it is relatively inexpensive and simple to use. It is especially effective with small databases but has limited use with large databases with numerous "fraud symptoms" that take unnecessary time and costs to investigate [Albrecht, 2003].

Another inductive method to search for symptoms of fraud in databases is with the use of Benford's law. This method looks for unusual patterns or anomalies in information in various types of data sets. It accurately predicts for most kinds of financial numbers that the first digit in a set of numbers will be very similar to a distribution pattern developed by Benford. This method analyzes the frequency of digits in a sample of numbers and assumes for the first digit in a particular number that the digit 1 occurs more frequently than the digit 2 and the digit 2 will occur more often then the digit 3 and so on. Benford's distribution pattern indicates that the digit 1 will be expected as the first digit about 30% of the time whereas the digit 9 will be expected about 4.6% of the time. This method does not apply to assigned numbers like SSN [Albrecht, 2003], but can be applied to dollar amounts if they satisfy some requirements.

To illustrate Benford's law, let's go back to the kickback fraud example mentioned above. To search for unusual patterns, we could take the invoices from each vendor and compare the fre-

quency of the first digit in the dollar amount on each invoice to the distribution pattern established by Benford. If the pattern found for a particular vendor differs from Benford's frequency pattern, then this could be considered unusual and a symptom that warrants further investigation.

An advantage of Benford's law is that it is relatively inexpensive, easy to implement, and can be applied to large databases. One disadvantage is that it takes a relatively broad rather than narrow approach to detecting fraud. As a result, a lot of false signals may occur, and if so, could be time consuming and costly to investigate. Similarly to ACL, this approach does not point to *the individual suspicious case*, but rather to a broad set of the possible fraud locations where the fraudulent transaction(s) should be found.

There are many indicators of possible suspicious (abnormal) transactions in traditional illegal business. These include (1) the use of several related and/or unrelated accounts before money is moved offshore, (2) a lack of account holder concern with commissions and fees [Vangel, James, 2002], (3) correspondent banking transactions to offshore shell banks [Vangel, James, 2002], (4) transferor insolvency after the transfer or insolvency at the time of transfer, (5) wire transfers to new places [Chabrow, 2002], (6) transactions without identifiable business purposes, and (7) transfers for less than reasonably equivalent value. Some of these indicators can be and actually implemented as simple flags in software. However, indicators such as wire transfers to new places produce a large number of 'false positive' suspicious transactions. Thus, the goal is to develop more sophisticated detection routines based on interrelations among many indicators.

Data mining approach

Recently data mining methods attracted attention for solving security and criminal detection problems [Thuraisingham, 2003; Mena, 2003]. Mena [2003] reviewed the subject (intelligent agents, link analysis, text mining, decision trees, self-organizing maps, machine learning, and neural networks) for security managers, law enforcement investigators, counter-intelligence agents, fraud specialists, and information security analysts. Brause, Langsdorf and Hepp [1999] discuss importance of use of non-numeric data in fraud detection.

Data mining has two quite different meanings: a technical and a common sense meaning. In the common sense meaning, every method that assists in finding hidden patters in large data sets is a data-mining method. *Technical data mining* is typically associated with (a) *supervised learning* based on training data of known fraud and legitimate cases and (b) *unsupervised*

learning with data that are not labeled to be fraud or legitimate. Bedford's law can be interpreted as an example of unsupervised learning [Bolton, Hand, 2002].

Technical data mining has a long history and a large variety of methods and software systems. However, the direct application of these methods to forensic accounting is limited due to almost complete non-existence of large sets of fraud training data [Jensen, 1997; Bolton, Hand, 2002]. In practical financial situations fraud cases are *rare events* relative to the total number of financial transactions. Thus, the fraud detection problems is a special type of data mining problem. These types of problems are known as **problems with rare and unbalanced patterns** (number of fraud training cases is much smaller than the number of or normal cases) that is only now starting to gain attention [Weiss, 2004; Lin, Chalupsky, 2003; Rattigan, Jensen, 2005; Getoor, 2003; Badia, Kantardzic, 2005].

Data mining for such rare and imbalanced patterns, along with link discovery, have both recently became important areas of research to meet two closely connected challenges: (1) the automated discovery of financial fraud and (2) the automatic linking of disparate information to help to fight terrorism. In this paper we propose a data-mining method that can deal with rare and unbalanced patterns in a variety of financial data.

The following data illustrate the size of datasets in some fraud detection problems: 350 million credit card transactions a year by Barclaycard in the United Kingdom; over a billion transactions a year by RBC, and around 275 million calls each weekday carried by AT&T [Bolton, Hand, 2002]. The same authors referenced several publications that report 0.1- 0.5% of fraudulent cases out of all wire and credit card transactions. This imbalance also is reported globally (approximately 10 million fraudulent transactions out of some 12 billion credit card transactions made annually [Hassibi, 2000].)

Such numbers make it obvious that the traditional *misclassification rate* is not an appropriate performance measure because we can get a misclassification error rate of only 0.001 with just classifying every transaction as legitimate [Jensen, 1997; Bolton, Hand, 2002]. In addition, two significant specific obstacles for building fraud detection models are the lack of labeled cases and intelligent adversaries that are highly adaptive and creative [Jensen, 1997].

Data mining can assist in discovering patterns of fraudulent activities, including those related to terrorism, such as transactions without identifiable business purposes. The fundamental problem is that an individual transaction often does not reveal that it has no identifiable business purpose, or that it was done for no reasonably equivalent value. Therefore, data mining

techniques must search for suspicious patterns in the form of *more complex combinations of transactions* and other evidence using background knowledge. Also, in this case, the training data are formed not by transactions themselves, but by combinations of two, three, or more transactions. This implies an explosion in the number of training objects. The percentage of suspicious records in the transaction set is very small, but the percentage of suspicious combinations in the set of combinations is minuscule. This is a typical task of *discovering rare patterns*. Traditional data mining methods are ill-equipped to deal with such problems. Relational data mining methods [Kovalerchuk, Vityaev, 2000, 2005] open new opportunities for solving these tasks by discovering "negated patterns" or "negative patterns," as described below.

After tools such as ACL are run, an auditor still needs to find individual suspicious transactions associated with fraud to prove fraud and make a legal case. ACL allows us to search for patterns that already have been *defined by humans* (such as "increasing purchase volume from one vendor when the total purchase volume is not growing"). It is not necessary that this pattern exists in a particular database, but it is clear what we are searching for. This means that the ACL process has two steps: (1) defining and recording patterns manually using the Audit Command Language and (2) testing pattern presence in the database automatically. This approach has limitations; creative people involved in fraud produce new fraud schemes every day and races with them by *manual pattern definitions* is not the best strategy in the long run.

A relational data mining approach can avoid manual pattern definition and accomplishes both steps automatically if training data are provided that have fraud and legitimate cases. The difficulty in forensic accounting and fraud analysis is that nobody provides enough (if any) fraud cases to the auditor in the audited company. Even if such an unlikely event happens, the training data may include only a few records from previous audits. On the other hand, the traditional data mining approach needs massive training data records to produce reliable formalized patterns using inductive data mining algorithms, such as neural networks and decision trees. Outside of audit problems this massive data requirement is often satisfied easily, e.g., in optical character recognition we can provide millions of training characters for each letter and discover reliable patterns. Provost [2000] suggested looking to methods for "profiling" the majority class without reference to instances of the minority class for imbalanced data when there simply are too few data to represent certain important aspects of the minority class. We are developing this approach further in this paper.

Discovering sparse evidence contained in large amounts of data sources is a new area of Data Mining called Link Discovery (LD). Currently LD mostly relies on deterministic graphical techniques. Bayesian probabilistic and causal networks are other relevant techniques.

Correlation of complex evidence that involves structured objects; text and data in a variety of discrete and continuous scales (nominal, order, absolute and so on) require development of a new technique that needs to meet a variety of requirements. One of them is model comprehensibility [Pazzani, 2000]. Complexity of this task dictates the need to combine the variety of mathematical techniques.

METHOD

Steps

We are expanding capabilities of data mining methods by defining fraud patterns automatically by using a new **Hybrid Evidence Correlation (HEC) method**. Bellow we informally outline steps of this method that are followed by its justification and a more exact description:

- Step 1: Assembling a new database of pairs of transactions from several databases (it can be databases of companies that do business with each other including banks that handle transactions).

- Step 2: Discovering automatically most frequent data association patterns in the new database *without using any fraud training data*, but with possible use of *a fraud ontology* (a structured set of concepts of fraud domain with relations between them). Tips from auditors on possible classes of fraud patterns (not individual patterns that are of too great a variety) can be used at this stage too.

- Step 3: Negating patterns from step 2 to get fraud candidate patterns (*negative patterns*). To get negative (negated) patterns for patterns in the form of if-then rules, we negate *only the then-part* of the rule. For instance, we may discover a frequent rule (pattern) that involve conditions A_1, A_2,…,A_{n-1} and conclusion A_n such that:

  If conditions A_1 & A_2 &…A_{n-1} are true – Then conclusion A_n is true.

  The rule that defines a candidate for the fraud pattern will be *a negative rule* where *only the conclusion* A_n is negated:

  If conditions A_1 & A_2 &…A_{n-1} are true – Then conclusion A_n is false.

- Step 4: Searching pairs of database transactions that satisfy a negative rule.

- Step 5: Providing cases (pairs of transactions) from step 4 to the auditor as suspicious for further investigation. At this step an auditor needs to separate random error in database from really suspicious patterns. One of the criteria for separation can be the *distribution* of suspicious cases between different rules from found in step 3. If the distribution is relatively even then it is likely that we have random errors or a very sophisticated fraud that was deliberately spread. If most of the cases are concentrated in a few rules then it is likely that we have a specific type of fraud or a systematic error (that is not random). The next development of this step will be a more general approach to *conducting a full scale second data mining process* on identified suspicious cases to separate random errors from really suspicious cases and systematic errors. The advantage of the second data mining can be that we will have a much more manageable dataset than the whole set of transactions. The same approach can be applied to combinations of transactions that include more transactions than only pairs of them.

For steps (1) and (2) a forensic accounting expert provides a class of potential patterns that will be explored automatically. This is in contrast with the ACL manual approach. The HEC method also differs from the traditional unsupervised pattern learning (clustering) technique known in data mining to find suspicious patterns as outliers. The HEC method provides patterns that can be easily understood by an auditor. Standard clustering techniques group data in clusters, and for a user to provide meaning for these clusters is often difficult or even impossible.

Consider a dataset of transactions records $R=\{r_i\}$ with attributes such as seller, buyer, item sold, item type, amount, cost, date, company name, and company type. Records r_1 and r_2 are *linked records* if the buyer B_1 in the first transaction r_1 is a seller S_2 in the second transaction r_2, that is, $B_1=S_2$. It is also possible that the item sold in both records is the same $I_1=I_2$.

We create a new dataset of all pairs of linked records (R_i,R_j) with $B_i=S_j$ and $I_i=I_j$. It is possible that *some definitions of normal and suspicious patterns* are provided such as listed below:

- a normal pattern (NP) – a Manufacturer Buys a Precursor, includes it in the manufacture of a finished product and Sells the finished product (Result) (MBPSR);

- a suspicious (abnormal) pattern (SP) – a Manufacturer Buys a Precursor and Sells the same Precursor (MBPSP);

- a suspicious pattern (SP) – a Trading Co. Buys a Precursor and Sells the same Precursor Cheaper (TBPSPC);

- a normal pattern (NP) – a Conglomerate Buys a Precursor, includes it in the manufacture of a finished product and Sells the finished product (Result) (CBPSR).

If definitions of *suspicious patterns* are given, then finding *suspicious records* is a matter of performing a computationally efficient search in a database (or distributed databases, which can be challenging too, but is not the subject of this paper). These definitions can be programmed in systems like ACL to find and separate suspicious and normal cases. The algorithm A may analyze all linked pairs of records (R_i,R_j) with, say 18 attributes total, and can match a pair (#5,#6) of records r_5 and r_6 with a normal pattern MBPSR, A(#5,#6) = MBPSR, while another pair (#1,#3) of records r_1 and r_3 can be matched with a suspicious pattern, A(#1,#3) = MBPSP. However ACL is not able to generate automatically (i.e., *to discover*) the *definition of pattern* MBPSP from a dataset of pairs of linked records (R_i, R_j).

Definitions of suspicious patterns, such as MBPSP (Manufacturer Buys a Precursor and Sells the same Precursor) can be recorded in the form of *if-then rules*: MBP ⇒ SP. Similarly, a normal pattern MBPSR (Manufacturer Buys a Precursor and Sells the Result of manufacturing) can be written as MBP ⇒ SR.

Data mining methods may be able to discover these definitions in the form of rules or associations, but they need a large number of MBPSP cases along with non MBPSP cases provided for training the data-mining model. The association rule method [Agrawal et al., 1993] fits well to finding such associations if training data are available.

Our goal is *to discover* (1) **rare pattern definitions (RPD)** and (2) **suspicious patterns (SP)** among rare patterns when there is no sufficient training data. One can ask: "Why do we need to discover these definitions automatically?" A manual way can work if the number of types of suspicious patterns is small and an expert is available. For multistage money-laundering transactions, this is difficult to accomplish manually. Creative criminals and terrorists are constantly inventing new and more sophisticated money laundering and other fraud schemes. There are, therefore, no statistics to learn as it is done in traditional data mining approaches.

An approach based on the idea of "negated patterns" can uncover such unique schemes. In this approach at step 2 *highly probable patterns* are discovered first and then their conclusion (then-part) is negated. It is assumed that a highly probable pattern should be *normal*. In more formal terms, the main hypothesis (MH) of this approach is: *If Q is a highly probable pattern (e.g., >0.9) then Q constitutes a normal pattern and Q with negated conclusion can constitute a suspicious (abnormal) pattern.*

We use the **probabilistic semantic inference (PSI)** for automatic generation of patterns. It is proved in [Vityaev E.E, 1992] that PSI provides all patterns with the highest values of conditional probability. These patterns are most reliable that can be found of a given dataset. Other data mining methods that search for *association rules* can find good patterns too, but they may not find all most reliable normal and abnormal patterns, because they use different pattern search criteria. Thus, theoretically probabilistic semantic inference can solve the problem. One of the goals of this paper is to show this is not only a theoretical possibility. Computational experiments with two synthesized databases and some suspicious transactions schemes permitted us to discover suspicious transactions.

The actual relational data-mining algorithm used was a computationally efficient algorithm MMRD (Machine Method for Discovery Regularities) that includes PSI [Kovalerchuk, Vityaev, 2000; Vityaev, 1992]. This algorithm is based on both first-order logic (FOL) and PSI. This algorithm is a result of the research in deterministic and probabilistic FOL inductive methods started in 1970s in Russia. This research includes a fundamental theorem proved by Samokhvalov [1973], several dissertations and conference proceedings, e.g., MOZ'76 [Zagoruiko, Elkina, eds., 1976].

Currently many methods based on FOL are developed and known under different names such as Inductive Logic Programming (ILP) methods, probabilistic and stochastic ILP [Dzeroski, 1996; Puech, Muggleton, 2003; Muggleton, 2002], probabilistic relational models [Dzeroski, Lavrac, 2001;Getoor et al, 2001], first order Bayesian networks], first order decision trees and relational probability trees [Provost, Domingos; 2003, Neville et al, 2003]. Learning from relational data is a subject of several recent workshops and conferences such as AAAI, IJCAI, MRDM, and KDD.

Advantage of using semantic probabilistic inference and First Order Logic

Use of first order logic (FOL) rules has an important advantage – a larger and deeper set of rules can be discovered than using other methods [Puech, Muggleton, 2003, Dzeroski, 1996;

Kovalerchuk, Vityaev, 2000]. This is important in fraud detection to be sure that critical rules are not missed because a limited data mining language does not allow discovering the rule.

The first-order logic and probabilistic semantic inference allows one to deal with noisy data and to express in essence *any fraud scheme* with *any data types* involved that can be numeric or non numeric. More traditional data mining methods are often limited to particular and well-known data types such as numeric data. Traditional methods have difficulties handling data types where each individual data unit is a structure not a single number, e.g., graphs of social relations. FOL methods (also known as relational methods) deal with such complex data types by transferring all interpretable information that data types contain into the expressions in the FOL language. Many specific techniques for such transfer have been developed in the Representative Measurement Theory [Krantz at al, 1971, 1980, 1990] that was started at Stanford University to meet challenges of handling a wide variety of data types that are abound in psychology.

Details of HEC technology

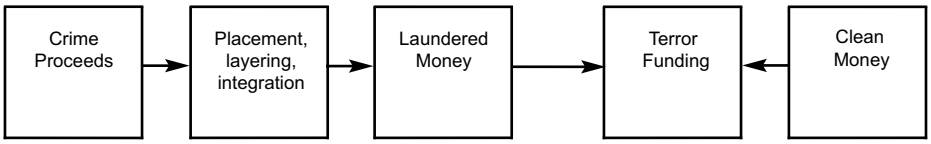
To be able to use HEC core steps 1-5 described above, several HEC preprocessing steps should be performed:

- Step Pr1. Identify the class of events of interest for searching (money transfer, deposit, payment on account and others) and concepts of normal and anomalous patterns for these events. This does not end up with a complete definition of suspicious patterns, but narrows its class for further discovery.

- Step Pr2. Identify a vocabulary that describes events of interest (more generally it means building a domain ontology).

Example: A money-laundering expert describes a terrorist related scheme, shown in Figure 1.

Figure 1
Money laundering and terrorism (based on Prentice, 2002)



The expert formulates the main properties of the scheme without much detail. Next the expert picks an individual event (case) and records it in the vocabulary he/she identified and generalizes this case by substituting certain predicates, object names and constants by variables and more predicates (properties). For instance, having the case that if A(a) then B(a), (A(a) ⇒ B(a)) the expert can generalize it stating that for every x If A`(x) then B`(x), which can be written formally as ∀ A`(x) ⇒ B`(x). Predicates (properties) A` and B` can be clarified later by the HEC algorithm using A and B. Such user involvement allows the search to be narrowed for suspicious pattern definitions. Next, the HEC core steps 1-4 are performed starting with data collecting and organizing as outlined in step 1.

At the HEC step 2 all most probable patterns, are detected and those that have a high probability above a certain threshold, say 0.9 are selected. To get most probable patterns, PSI inference will add all additional features providing maximization of probability of the hypothesis. Thus, all possible more precise expressions of the given scheme/case will be found. It was proved in Vityaev [1992], that PSI reveals all rules with maximum possible values of conditional probabilities. This ensures the revealing all normal and all negated normal (anomalous) patterns that cover the given scheme/case.

Then HEC core steps 3-5 will provide suspicious pairs of records to the user for further investigation and the user will start **HEC post-processing steps**:

- Step Pp1. Performing expert analysis of obtained anomalous patterns. Choosing patterns that reflect the illegitimate actions.

- Step Pp2. Analyzing patterns of illegitimate actions. To this end, considering all events they describe.

Consider three situations and possible actions at step Pp2:

1. All events of the patterns are illegitimate. If so, the pattern of illegitimate actions is detected.
2. If some, not all, events of the anomalous pattern are illegitimate, compare these events with other events of the anomalous pattern, which are not illegitimate, and determine the additional features that make the legitimate and illegitimate patterns different. Using the detected features, formulate the more precise pattern for illegitimate actions.

3. If all events of the given anomalous pattern are, indeed, legitimate then find the more precise characterization of the legitimate pattern. Check that the more precise legitimate pattern has no anomalous cases (cases that satisfy its negated then-part). Otherwise, analyze the new anomalous pattern found.
4. Alternatively the additional features for steps 3 and 4 can be selected automatically by an algorithm if a list of candidate features is defined in advance and a data-mining algorithm is capable to work with small samples of illegitimate and anomalous cases.

This adaptive learning process will increase training data set and will make a system more and more useful in the course of using it.

HYBRID EVIDENCE CORRELATION (HEC) MODEL

General model components

The HEC model outlined above combines first-order logic (FOL) and probabilistic semantic inference (PSI) with the following main concepts involved:

- a set of evidences D described by a set of attributes {Atr} and relations (predicates, {Pr} with two or more arguments);
- a domain ontology in a variety of forms including a hieratical taxonomy of terms starting from {Atr} and {Pr} as terminal nodes;
- formalized definitions of normal patterns, {Norm} and suspicious patterns, {Susp} in terms of FOL and PSI;
- classification of patterns (statistically significant vs. insignificant ones to capture important rare events),
- a generator of potential suspicious patterns (hypotheses generator), G,
- an evaluator of hypotheses/patterns E and
- a selector of suspicious patterns L.

In the Testing Hypothesis section we shall present details of successful testing of the HEC approach for detecting fraud transactions on synthetic data that involved these components.

HEC Model for money laundering analysis

Discovering of suspicious patterns as defined in the Steps section can lead to discovering a kickback or actual money laundering that makes dirty money clean. It also can lead to discovering a terrorism link – a manufacturing/trading company could be used as a front company to hide actual buyer of the precursor (e.g., fertilizer) needed for making bombs. Similarly the fake Charity Foundation can be used to support terrorists and criminals.

Below we discuss how these patterns can be discovered automatically from an ordinary or distributed transactions database (DB). We assume that DB contains transactions with attributes such as: seller, buyer, item sold, amount, cost and date (see Table 1).

Table 1
Transaction records

Record ID	Seller	Buyer	Item sold	Amount	Cost	Date
1	Aaa	Ttt	Td	1t	$1000	03/05/99
2	Bbb	Ccc	Td	2t	$1000	04/06/98
3	Ttt	Qqq	Td	1t	$1000	05/05/99
4	Qqq	Ccc	Pd	1.5t	$1000	05/05/99
5	Ccc	Ddd	Td	2.0t	$2000	08/18/98
6	Ddd	Ccc	Pd	3.0t	$4000	09/18/98

Next, information about types of companies and items sold is also partially available (see Tables 2 and 3).

Table 2
Company types and Item types

Company name (seller/buyer)	Company type	Item	Item type in process
Aaa	Trading	Td	Precursor
Bbb	Unknown	Pd	Product
Ccc	Trading	Rd	Precursor
Ttt	Manufacturing	Td	Precursor
Ddd	Manufacturing	Pd	Product
Qqq	Conglomerate	Pd	Product

We need to assemble a new table (see Table 3) from tables 1-2 to reveal suspicious patterns in records. Neither table 1 or 2 indicate these individually. Table 3 also does not indicate suspicious patterns immediately. But we can map each pair of records in Table 3 to patterns listed above using a pattern-matching algorithm A that analyzes pairs of records in Table 3.

Table 3
Pairs of transactions

Record ID	Seller	Seller type	Buyer	Buyer type	Item sold	Item type	Amount	Price	Date
1	aaa	trading	Ttt	manuf.	Td	Precursor	1t	$1000	03/05/99
2	bbb	unknown	Ccc	trading	Td	Precursor	2t	$2003	04/06/98
3	ttt	manuf.	Qqq	Congl.	Td	Precursor	1t	$1000	05/05/99
4	qqq	Congl.	Ccc	trading	Pd	Product	1.5	$2000	06/23/99
5	ccc	Trading	Ddd	Manuf.	Td	Precursor	2.0	$2000	08/18/98
6	ddd	Manuf	Ccc	trading	Pd	Product	3.0	$4000	09/18/98

Thus we can map pairs of records in Table 4 into the following patterns:

- A(#5,#6)=MBPSR, a Manufacturer Buys a Precursor, includes it in the manufacture of a finished product and Sell the finished product (Result) (MBPSR);
- A(#1,#3)= MBPSP, that is a manufacturer bought a precursor and sold the same precursor (suspicious pattern);
- A(#2,#5)= TBPSPC, that is a trading Co. bought a precursor and sold the same precursor cheaper (suspicious pattern).

Now let us assume that we have a database of five billon transactions of the type presented in Table 3. Statistical computations can reveal a distribution of these pairs into patterns (i.e., frequencies) as shown in Table 4.

Table 4
Example of frequencies

Pattern	Type	%	Approximate number of cases
MBPSR	normal	55	$0.55*5*10^9$
MBPSP	suspicious	0.1	100
CBPSR	normal	44.7	$0.44*5*10^9$
TBPSPC	suspicious	0.2	200

Thus we have 100+200=300 suspicious transactions. This is 0.3% of total number of transactions and about $6*10^{-6}$ % of the total number of pairs analyzed. It shows that finding such transactions for large and distributed databases is a tremendous challenge, but an underlying algorithm A for each found pattern is relatively simple (see pseudo-code in Table 5). This is because we have only two suspicious pattern/hypotheses defined in advance.

Table 5
Algorithm for finding records that match suspicious patterns MBPSP and TBPSPC

1. Form an SQL-query (Q1) to DB to retrieve pair of records that satisfy MBPSP

   Make Table 3 from Tables 1 and 2;

   Make an SQL-query to find a pair (MBP record, and matching SP record) in Table 3.

2. Form an SQL-query (Q2) to DB to retrieve pairs of records that satisfy TBP-SPC;

   Use Table 3 from above;

   Make an SQL-query to find a pair (TBP record, and matching SPC record) in Table 3.

3. Run query Q1 in a DB;

4. Run query Q2 in a DB.

The number of potential normal and abnormal types of patterns can be much larger and **automatic generation of patterns/hypotheses descriptions** is a major challenge that we are addressing in this paper. Thus, our major question is: *How to generate automatically suspicious patterns/hypotheses using DBs*? This includes generating MBPSP and TBPSPC descriptions automatically. Here we do not assume that we already know that MBPSP and TBPSPC are suspicious. We have previously discussed the question: "Why do we need to discover these definitions (rules) automatically?" The answer was that a manual way can work if the number of types of suspicious patterns is small and an expert is available. For multistage money laundering transactions, and many complex theft manipulations, the first condition does not hold, and therefore it is difficult to accomplish this manually.

As we mentioned above, our approach to identify suspicious patterns is to discover highly probable patterns and negating them. We suppose that a highly probable pattern should be normal.

In more formal terms the main statement/hypothesis (MH) is:

If Q is a highly probable pattern (>0.9) then Q constitutes a normal pattern and conclusion-negated (Q) can constitute a suspicious (abnormal) pattern. (1)

Table 6 outlines an algorithm based on this hypothesis to find abnormal (suspicious) patterns. The algorithm applies first-order logic (FOL) and probabilistic semantic inference (PSI). See Vityaev [1992] for more mathematical detail and a theorem on computational efficiency.

Table 6

HEC algorithm steps for finding suspicious patterns based on Main Hypothesis

1.	Assemble a new database of pairs of transactions from several databases (it can be databases of companies that do business with each other including banks that handle transactions).
2.	Discover patterns as Horn clauses, $A_1\&A_2\&\ldots\&A_{n-1}\Rightarrow A_n$; e.g., MBP⇒SR. Generate a set of predicates **P**=$\{P_1,P_2,\ldots,P_m\}$ and first order logic (FOL) literals $A_1,A_2,\ldots,A_n$ based on **P**. Compute a probability $P(A_1\&A_2\&\ldots\&A_{n-1}\Rightarrow A_n)$ such that $A_1\&A_2\&\ldots\&A_{n-1}\Rightarrow A_n$ is true on a given database. This probability is computed as a conditional relative frequency. $P(A_n/A_1\&A_2\&\ldots\&A_{n-1})=N(A_n/A_1\&A_2\&\ldots\&A_{n-1})/N(A_1\&A_2\&\ldots\&A_{n-1})$, where $N(A_n/A_1\&A_2\&\ldots\&A_{n-1})$ is the number of cases with A_n being true when $A_1\&A_2\&\ldots\&A_{n-1}$ are true, and, $N(A_1\&A_2\&\ldots\&A_{n-1})$ is the total number of $A_1\&A_2\&\ldots\&A_{n-1}$ cases $N(A_1\&A_2\&\ldots\&A_{n-1})$. Compare $P(A_1\&A_2\&\ldots\&A_{n-1}\Rightarrow A_n)$ with a threshold T, e.g., T=0.9. If $P(A_1\&A_2\&\ldots\&A_{n-1}\Rightarrow A_n)>T$ then a database is 'normal," e.g., P(MBP⇒SR) can be 0.998 Test statistical significance of $P(A_1\&A_2\&\ldots\&A_{n-1}\Rightarrow A_n)$. We use Fisher criterion (for more detail see [Kovalerchuk, Vityaev, 2000] to test statistical significance.
3.	If the database is "normal" $P(A_1\&A_2\&\ldots\&A_{n-1}\Rightarrow A_n)>T=0.9$ and rule R: $A_1\&A_2\&\ldots\&A_{n-1}\Rightarrow A_n$ is statistically significant, then negate the then-part of R to produce a new rule: $A_1\&A_2\&\ldots\&A_{n-1}\Rightarrow \neg A_n$. Compute probability $P(A_1\&A_2\&\ldots\&A_{n-1}\Rightarrow \neg A_n) = 1 - P(A_1\&A_2\&\ldots\&A_{n-1}\Rightarrow A_n)$. In the example above it is 1-0.998=0.002.
4.	Searching pairs of transactions in the database that satisfy rules with the negated then-part, $A_1\&A_2\&\ldots\&A_{n-1}\Rightarrow \neg A_n$
5.	Analyze database records that satisfy $A_1\&A_2\&\ldots\&A_{n-1}\&\neg A_n$ to identify real fraud cases.

To minimize computations we can generate randomly a part of all possible pairs of records such as is shown in Table 3. Then, an algorithm finds highly probable (P>T) Horn clauses. Next, conclusions of these clauses are negated. After that a full search of records in the DB is performed to find records that satisfy negated clauses. According to our main hypothesis (1), this set of records will contain suspicious records and search of actually "red flag" transactions will be significantly narrowed.

Use of the property of *monotonicity* is another way to minimize computations. We may discover that conditions A and B and conclusion C are sufficient to recognize the pattern A&B ⇒ C as suspicious. In other words, it does not matter if any other possible condition D or its

negation ¬ D is satisfied too. This pattern is suspicious independently if D is true or false, that is both patterns A & B & D ⇒ C and A & B & ¬ D ⇒ C will be suspicious too. This means that we can save time avoiding testing both patterns A & B & D ⇒ C and A & B & ¬ D ⇒ C if we already know A&B ⇒ C. This approach was successfully used in other domains [Kovalerchuk, Vityaev, 2000, 2001].

Testing hypothesis

To test HEC approach and our Main Hypothesis (1) we designed two test experiments:

1) Test 1: Generate a relatively large syntactic database (an extended Table 3) that includes a few suspicious records MBPSP and TBPSPC. Run the HEC algorithm to discover as many highly probable patterns as possible. Check that patterns MBPSR and CBPSR are discovered among them. Negate MBPSR and CBPSR to produce patterns MBPSP and TBPSPC. Run patterns MBPSP and TBPSPC against the DB to find all suspicious records consistent with them.

2) Test 2: Check that other found highly probable patterns are normal and check that their negations are suspicious patterns (or contain suspicious patterns).

A positive result of test 1 may confirm our hypothesis (1) for MBPSR and CBPSR and their negations. Test 2 may confirm a wider set of patterns. A test 1 method contains several steps:

- Create a Horn clause: MBP ⇒ SR.
- Compute a probability that MBP ⇒ SR is true on a given database. Probability P(MBP ⇒ SR) is computed as a conditional probability P(SR/MBP)= N(SR&MBP)/N(MBP), where N(SR&MBP) is the number of MBPSR cases and N(MBP) is the number of MBP cases.
- Compare P(MBP ⇒ SR) with 0.9. If P(MBP ⇒ SR)>0.9 then a database is 'normal."
- Test statistical significance of P(MBP ⇒ SR) using the Fisher criterion.
- If the database is "normal" (P(MBP ⇒ SR) >T=0.9) and P(MBP ⇒ SR) is statistically significant then negate then-part of MBP ⇒ SR to produce MBP ⇒ ¬ SR. Threshold T can have another value too.
- Compute probability P(MBP ⇒ ¬ SR) = 1- P(MBP ⇒ SR).
- Analyze the database records that satisfy MBP and ¬ SR.

Thus, if probability P(SR/MBP) is high and statistically significant then we can say that a normal pattern MBPSR is discovered. Then we suppose that suspicious cases are among cases where MBP is true but conclusion SR is not true. We can collect all such cases and start to analyze the actual content of the then-part of the clause MBP ⇒ SR. We can discover that the set of cases with ¬ SR contains a variety of entities. Some of them can be very legitimate cases. Therefore, this approach does not guarantee that we find only suspicious cases, but the method narrows the search to a much smaller set of records.

EXPERIMENT

We generated a synthetic database with attributes shown in Table 3. It contains data that satisfy normal patterns with some exceptions, e.g., MBP ⇒ SR is true only in about 95% of the cases. For some cases we have that a manufacturer bought a precursor and sold the precursor not a product, MBP ⇒ SP. Using a HEC algorithm we were able to discover this pattern and other highly probable patterns. The HEC approach is implemented as MMDR algorithm (see pseudo-code in [Kovalerchuk, Vityaev, 2000]). It worked without any information in advance that these patterns are in data. In our computational experiments the total number of patterns discovered is 41. The number of triples of companies (i.e., pairs of transactions) captured by the patterns is 1531 out of total 2772 triples generated in the experiment. Table 7 depicts two statistically significant normal patterns with the following notation: Second_Buyer_type means a buyer in the second transaction, i.e., having A sold some item to B and B sold to C then C will be a Second_Buyer. Similarly, Second_Item means the item sold by B to C.

Table 7
Computational experiment: some discovered regularities, patterns

#	**Discovered regularity**	**Frequency**
1	IF New_Buyer__type = Manufacturing AND Item_type= precursor THEN New_Item_type = product A statistically significant normal pattern with P>0.9. It indicates that a Manufacturer (M) Bought (B) a precursor (P) and sold a product, MBP ⇒ SR. It is exactly the same normal pattern MBP ⇒ SR that was identified manually and now was discovered automatically. The negation of this pattern MBP ⇒ not (SR) is suspicious – a manufacturer bought a precursor, but did not sell a product (a result of manufacturing). The manufacturer could sell something different or sell noting. This happened in 8 cases that are suspicious and need to be examined in detail.	173/(8+173)= **0.955801**
2	IF Seller_type = Trading AND New_Buyer_type = Manufacturing THEN New_Item_type = product A statistically significant normal pattern with P>0.9. It indicates that a Manufacturer (M) Bought (B) something from a Trading (T) company and Sold (S) a product (R), MBT ⇒ SR. It fits the conclusion of the normal pattern MBP ⇒ SR, but it does not indicate that manufacturer M bought a precursor (P). However, the negation of this pattern MBT ⇒ not(SR) is suspicious – a manufacturer M did not sell a product of manufacturing. This is true for 4 cases that should be explored as suspicious.	99/(4+99)= **0.961165**

HOW IS HEC METHOD RELATED TO THE ASSOCIATION RULE METHOD?

The original association rule method [Agrawal et al, 1993] generalizes data in the form of a propositional rule

A & B &… G ⇒ Q.

The HEC method provides a two-step generalization:

1. A & B &… G ⇒ Q,

2. A & B &… G ⇒ not Q, and not Q ⇒ S,

where S is a suspicious situation. The first generalization discovers frequent patterns and the second step attempts to find rare patterns. Also HEC discovers the first order logic rules predicates [Mitchell, 1997, Flach, Lachiche, 2001] that can be more general than the original association rules method discovers [Agrawal et al, 1993]. More exactly, typically association rules can discover a relation A(x) & B(x) &… G(x) ⇒ Q(x) with a high level T of probability P of Q(x), when A(x) & B(x) &… G(x)) is true,

$$P(Q(x) /A(x) \& B(x) \&\ldots G(x)) > T \quad (2)$$

Here T is a threshold of the conditional probability P. In other words, association rules operate within the logic of monadic predicates that have only one argument x, A(x), B(x) and so on. This logic is equivalent to propositional logic [Mitchell, 1997, Flach, Lachiche, 2001]. Uncovering unlawful activities such as fraud schemes may need more complex relations than association rules support. For instance, we may need to discover a relation with predicates with more than one argument:

$$A(x,y) \& B(y,z) \&\ldots G(x,z,w) \Rightarrow Q(x,w), \quad (3)$$

Such type of relations seems natural in financial transactions analysis. Let x,y,z, and w be transactions and predicates A, B,… G. Q specifies relations between these transactions. For instance, the target predicate Q(z,w) can mean that transactions z and w form a kickback scheme, that is z is a base purchase transaction and w is a kickback payment transaction based on z but disguised via some intermediate transactions y, z and others. Relations A,B,… G between transactions uncover how kickback was implemented. Relation A can be a combination of two off-shore transactions and relation B can be a relations between three transactions

done by front companies. Equation (3) is written in the first order logic that is more general than used in equation (2). Developing of first order association rules as a part of unsupervised learning is a growing area of research and applications [Flach, Lachiche, 2001].

Below we summarize the important differences of the HEC technology based on the MMDR algorithm [Kovalerchuk, Vityaev, 2000] in comparison with association rule algorithms [Agrawal et al, 1993]. The differences in the set of rules produced follows from the differences in rule selection criteria.

1. At first we consider a *deterministic* situation when data have no noise and no item with the same attributes and properties belongs to different classes. In this deterministic situation the difference is that MMDR algorithm finds only one rule A&B ⇒ C, that is true on data, but the association rule algorithm finds also rules that are derived from this rule A&B ⇒ C by adding any additional condition D,F,... to the if part of the rule, i.e., A&B&D ⇒ C, A&B&F ⇒ C, ...

2. Having noise in data and overlapping classes (*non-deterministic* situation) the MMDR algorithm finds one rule A&B ⇒ C, which represents a statistical law with some level of statistical significance. The association rules algorithm finds **all** "specifications" of this rule such as A&B&D ⇒ C, A&B&F ⇒ C and so on that are deterministic and can forecast C.

3. Due to these differences the MMDR algorithm has predictive capability based on simplicity and statistical significance and can be used for prediction. This may be the case only for few rules that the association rule algorithm discovers, but the majority of the association rules may not have such predictive capabilities. They can suffer from the well-known overfitting problem that can be illustrated with an interpolation example. Having 100 points (x,y) we can build a polynomial F(x) to interpolate y. If the power n of F(x) polynomial is 100, then we can get an exact interpolation of given data D, F(x)=y, but beyond that data this interpolation can provide much larger errors than lower power polynomials as many empirical research had shown.

HANDLING DATA OF DIFFERENT TYPES

To uncover fraud we often need to use data of very different types (numeric, nominal, ordered, graphs of social relations and other structures). Each data type is characterized by a specific set of meaningful relations that can be used as a backbone of pattern discovery. The use of the rich language of the first order logic allows us to capture and manipulate such relations in its full extend. This is a subject of the representative measurement [Krantz et al, 1971, 1980,

1990]. As a result the HEC method opens the enormous possibilities of capturing fraud schemes using a variety of very different data types and relations in compliance with the Measurement Theory.

Example: We may wish to add new relations to formula (3) above to uncover a dipper and more specific fraud pattern. The HEC algorithm can uniformly and automatically do this if these relations are presented in the data type definition. Say, we want to add relations Cost Greater(x,y) and Next(x,y), where Cost Greater(x,y) is true if cost of transaction x in greater than cost of transaction y and Next(x,y) is true if transaction y follows the transaction x in time. In this example, the complex date type is a *transaction* data type that we view as is defined by several values and relations.

DETECTING ACTUAL FRAUDULENT TRANSACTIONS VS. COMPUTING GENERALIZED FRAUD INDEXES

From a financial viewpoint, the important advantage of the HEC method is in use of actual transactions instead of generalized indexes used in corporate fraud detection such as day's sales in receivable index, Gross Margin index, and asset quality index [Grove and Cook, 2004]. These indexes do not identify actual fraud and may miss some large fraud. For instance, Grove and Cook [2004] analyze Enron's, WorldCom, Global Crossing and Qwest financial reports and concluded that only Enron's indexes indicate a red flag. The HEC approach operates with actual transactions, which underlie the financial reports that are used to compute indexes. Thus, the HEC has a much greater chance to *discover* specific fraudulent transactions by identifying those in need of further analysis.

Benford's Law [Durtschi, Hillison, Pacini, 2004] is based on the number of times the particular digit occurs in a particular position in numbers. However, this method again does not reach a bottom line of individual suspicious transactions, in that it fails to narrow possibilities to a manageable set of promising leads [Albrecht, 2003] or to identify a perpetrator.

FUTURE WORK AND APPLICATION TO FINANCIAL FORENSIC SERVICES

The HEC data mining approach can be used in many types of forensic services within the financial arena, similar to those we have shown above. These services may include fraud risk assessments; background checks; information security risk assessment, asset tracing, end-user monitoring, vendor monitoring, and money laundering compliance program. A particular area

where the HEC approach can be useful is uncovering financial support of terrorism. Traditionally, detection of money laundering has been focused on the tracing of extensive operations with ready money. Today, the focus is shifting from seamy business and drug trade to terrorism. As aptly observed, "Terror funding presents an even greater challenge to the financial system since it can comprise both laundered money and clean money" [FSO, 2002]. In economic terms, illegitimate business and drug trade does merely "laundering dirty money," whereas terrorism rotate, along with laundering, "clean" money (for example, philanthropists funding). The following scheme makes apparent the distinction (see Figure 1), where (1) placement recaps starting funds obtained by illegitimate means, (2) layering isolates of illegitimate placements from their sources through multileveled intricate financial transactions and (3) integration imparts imaginary legitimacy to wealth amassed by criminal ways [FSO, 2002]. Current (since Sept. 11) practice includes cross-checking customer and account lists against names that appear on law enforcement lists of suspected terrorists and money launderers, names that are transliterated and thus spelled in multiple way [Vangel, James, 2002]. Data mining technology opens a way to enhance this process.

DATA ACQUISITION AND ONTOLOGY BUILDING

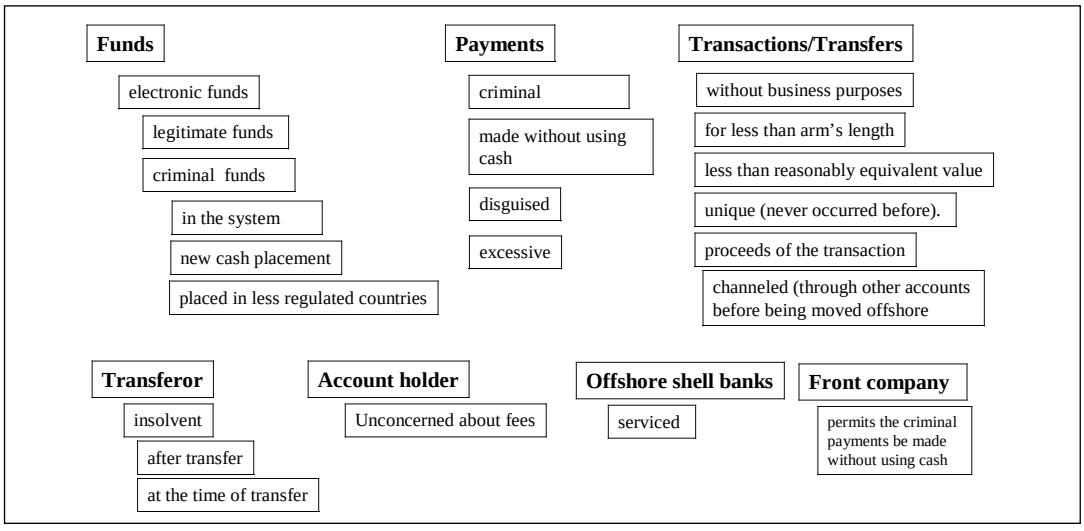
The HEC approach will be much more efficient if combined with an elaborated fraud ontology (a set of fraud area concept and patterns with their relationships), because sophisticated rules can use concepts from such ontology to discover fraud. At the best of our knowledge this ontology does not exist yet. Below we provide a few fraud concepts based on Prentice [2002], Vangel, James [2002], and other sources as a base for the fraud ontology:

- electronic funds (legitimate or criminal)
- criminal funds in the system
- criminal funds as new cash placement;
- criminal funds placed in the less regulated countries
- criminal funds wired into the better regulated jurisdictions
- front company (company that permits non-cash criminal payments)
- criminal payments
- disguised payment (e.g., payment for a shipment of drugs disguised as the delivery of non existent goods or services)

- excessive payment (payment for goods at a price in excess of the market price)
- channeled proceeds (proceeds of the transaction are channeled through several related and/or unrelated accounts before being moved offshore)
- transaction without business purposes (a transaction that appears to have no identifiable business purposes)
- unconcerned account holder (the account holder seems unconcerned about commissions and fees, despite their impact on the economics of the underlying transaction)
- correspondent services to offshore shell banks
- transferor insolvent
- transfer for less than arm's length
- transfer less than reasonably equivalent value
- unique wire transfers (wire transfers where they never occurred before).

Figure 2 depicts these concepts in a more structural form.

Figure 2
Fragment of fraud ontology for the financial system



SELECTING RULES FOR FRAUD DETECTION: CASE OF CONFLICTING FRAUD FLAGS

Last and Kandel [1999] provide an example that illustrates the problem of selecting rules for fraud detection when rules conflict each other. This example is about uncovering *calling card fraud*, where a simple discovered pattern R_1 may be that a person never uses his / her card on weekends, and, then, a half-an-hour conversation with Honolulu may be suspected as a fraud. However, there may be another pattern R_2 saying that 90% of the same person calls to Honolulu.

The two rules themselves do not conflict. Absence of calls to Honolulu on weekends does not mean that there is no such call on weekdays. However, the use of these rules for fraud detection provides inconsistent fraud detection with possible false alarms or missed frauds. The first rule R_1 interprets a specific event as suspicious but the second one R_2 does not. How can we avoid this?

If we would search for the most specific (and statistically significant) rules that can be discovered on the available dataset then the most specific rule will be R_1&R_2, but not two separate rules R_1 and R_2 that are both statistically significant.

Rule 1: If Call(John) then NoWeekendCall(John) and $0.9 < P(Rule1) < 1.0$

Rule 2: If Call(John) then WeekdaysCallHonolulu(John) and $0.9 < P(Rule2) < 1.0$,

Here P is a probability of the rule.

Rule 1&2: If Call(John) then NoWeekendCall(John) or WeekdaysCallHonolulu(John) and $0.9 < P(R1\&R2) < 1$.

Negation of then-part of this rule is –

NegRule1&2: If Call(John) then not(NoWeekendCall(John) or WeekdaysCallHonolulu(John))

and $P(NegRule1\&2) > 1-0.9=0.1$.

In more traditional form this rule will be

NegRule1&2: If Call(John) then WeekendCall(John) & NoWeekdaysCallHonolulu(John))

The cases that satisfy this rule (calls on weekends and no calls to Honolulu in weekdays) are really suspicious. They contradict both patterns R_1 and R_2. This example explains the importance of our dual HEC requirement for rules to be *most specific* and *statistically significant*. To the best of our knowledge, other methods do not pursue this dual requirement explicitly.

RARE EVENTS: FRAUD, ERRORS AND BENIGN ANOMALIES

We will call rules discovered by the HEC method *positive rules* and we will call these rules with negated then-part *negative rules* or *negated rules*. To make positive rules truly representative of normal legitimate business practice we distinguish *typical* and *atypical* positive rules/patterns among statistically significant rules. We compute the support of each rule (the number of cases for which the rule is true) and put rules in the descending order relative to the support. The rules at the top are called typical business rules, and rules at the bottom are called atypical business rules.

Example: We may have a typical rule that covers 5000 cases and an atypical rule that covers only 50 cases. We will negate the first rule that represents a normal business practice and will not negate the second rule that should be directly analyzed by the auditor. The second rule can be a fraud pattern that reveals fraud modus operandi or be an indicator of the systematic errors in the database. Cases that satisfy found negative rules can indicate: (i) random or systematic errors in the database, (ii) benign anomalies, or (iii) fraud.

Most likely *benign anomalies* and *random errors* in the database do not follow any pattern, i.e., they can be spread relatively evenly between negative rules or do not appear in them at all (an extreme case of even distribution). This is the base of our first criterion to separate benign anomalies and random errors from fraud and systematic errors. Systematic errors and fraud cases may follow some patterns (fraud modus operandi for fraudulent companies). Thus, actual fraud cases and systematic errors that involve the same company may follow few specific negative rules, notR_i.

Accordingly our first criterion for separation of fraud and systematic errors from benign anomalies and random errors is finding for each suspicious case its negative rules (rules where the case is true) and checking that these rules is a small group (relative to all negative rules) and each of these negative rules is true for several cases, not only one case.

Now we need to describe a criterion how to separate fraud from systematic errors. We assume that systematic errors corrupt only a small fraction of all records of a particular company in the database, say less than 1%. Otherwise they would be found already. If few systematic

errors take place for the company that does only legitimate business then the majority of its uncorrupted records should satisfy many positive rules (rules of legitimate business). Thus, the criterion to separate fraud records from systematic errors is checking that other records of the same company satisfy the normal business rules. If this is the case then the particular record is more likely a systematic error then fraud and the flag "possible systematic database error" will be provided to the auditor.

This idea can be described in the following way. Let S be a case of transactions that involve companies C_1, C_2 and C_3. Let this case also satisfy a negative rule $negR_i(S)$. We check if there are positive rules, R_j that are true for the same companies. If there are many such rules R_j then it is less likely that S is a fraud case. Positive rules indicate a pattern of normal behavior of companies. Thus having many positive indicators about these companies may mean that case S in less likely a fraud case.

Example: A manufacturing company C_1 bought a precursor and sold it without any manufacturing. It can be a relatively normal business (not enough buyers for their products in the market) or an indicator of some violations including fraud. If many normal rules are true for C_1 then case S is less likely a fraud case. This means that the company follows many patterns of normal (standard) business practice. If there is no such positive rule/pattern (or only a few of them) then it is more likely that the company is involved in an illegal business and fraud.

This idea can be elaborated to different models. Below we consider the following model of the structure of items/transactions: there is an exemplary (typical) item that is in the center of distribution of items of the class for every class of items. Other items of the class are distributed around the exemplary item with attributes that are random deviations of attributes of the exemplary item. In a more complex case deviation can follows some specific distribution law. In this model large deviations can represent (1) a normal case that has a low probability, or (2) a suspicious case that is not normal. One can explore the distribution of large deviations and identify that distribution. Then one looks at the cases that are deviations even in the set of large deviations.

This approach leaves unanswered a situation where a sophisticated fraud was disguised as a random error. That is, each fraud satisfies only one negative rule having many modus operandi involved. To elaborate this situation and enhance the approach as whole a full-scale second level data mining process on identified suspicious cases can be built. The advantage of the second data mining can be that we will have a much more manageable dataset than the whole set of transactions. The same approach can be applied to combinations of transactions that include more transactions than only pairs of them. On the following pages we provide a numeric example that illustrates the approach.

Table 10
Data Example

Company	Buys	Price	Sells	Price
Company1	Monitor	$390	Computer	$900
	Processor	$75	System block	$500
	HDD	$96		
	Video	$50		
	CD drive	$64		
	Power	$53		
	Motherboard	$64		
Company 2	Monitor	$400	Computer	$950
	Processor	$80		
	HDD	$100		
	Video	$50		
	CD drive	$60		
	Power	$55		
	Motherboard	$66		
Company 3	Monitor	$385	Computer	$1000
	Processor	$77	System unit	$500
	HDD	$95		
	Video	$55		
	CD drive	$60		
	Power	$50		
	Motherboard	$65		
Company 4	Monitor	$385	TV	$500
	Processor	$77	System unit	$500
	HDD	$95		
	Video	$55		
	CD drive	$60		
	Power	$50		
	Motherboard	$65		
Company 5	Monitor	$385	Monitor	$485
	Processor	$77	Processor	$87
	HDD	$95	HDD	$105
	Video	$55	Video	$65
	CD drive	$60	CD drive	$70
	Power	$50	Power	$60
	Motherboard	$65	Motherboard	$75

Analyzing these companies we discover:

Rule 1 (RU1):
If Manufacturer (Buy Monitor) & (Buy Processor) & (Buy HDD) & (Buy Video) & (Buy CD drive) & (Buy Power) & (Buy Motherboard) – Then it Sells Computer with probability 0.99 and sells System Units with probability 0.6; and Sells only System Units with TV (using monitor) without computers with probability 0.1

Rule 2 (RU2):
If Manufacturer (Buy Monitor) &(Buy Processor) & (Buy HDD) & (Buy Video) & (Buy CD drive) & (Buy Power) & (Buy Motherboard) – Then it Sells System Unit with probability 0.6.

The last rule is weaker because it has a lower probability.

The negative rule can be:

Rule 3 (RU3):
If Manufacturer (Buy Monitor) & (Buy Processor) & (Buy HDD) & (Buy Video) & (Buy CD drive) & (Buy Power) & (Buy Motherboard) – Then it does not (Sell Computer).

RU3 may overlap with RU2 for system units. Say 10% of companies make only system units, with TVs or Monitors as tools for browsing of pictures instead of computers. To avoid false alarm these cases need to be excluded from RU3. We add negated conclusion of RU1 to the if-part of RU3 and we get a more specific if-part of the negative rule:

Manufacturer (Buy Monitor) & (Buy Processor) & (Buy HDD) & (Buy Video) & (Buy CD drive) & (Buy Power) & (Buy Motherboard) & not (Sell Computer) & not (Sell System Blocks).

Trading company #5 satisfies this rule.

At the same time, a whole group of other rules (presented below) are violated for this company, which illustrates an idea of *analysis of violation of a group of rules*.

Negative rules that are violated for the trading company #5 are:

Company (Buy Monitor) AND Sell(Monitor)

Company (Buy Processor) AND Sell(Processor)

Company (Buy HDD) AND Sell(HDD)

Company (Buy Video) AND Sell(Video)

Company (Buy CD drive) AND Sell(CD drive)

Company (Buy Motherboard) AND Sell(Motherboard)

Company (Buy Power) AND Sell(Power).

These rules are produced by negating then-part of the following rules:

Company (Buy Monitor) THEN not Sell(Monitor)

Company (Buy Processor) THEN not Sell(Processor)

Company (Buy HDD) THEN not Sell(HDD)

Company (Buy Video) THEN not Sell(Video)

Company (Buy CD drive) THEN not Sell(CD drive)

Company (Buy Motherboard) THEN not Sell(Motherboard)

Company (Buy Power) THEN not Sell(Power).

The filtering we have described herein can be viewed as part of the general area called discovering *interesting patterns* found by using association rules. Badia and Kantardzic [2005] noted that most association rule mining algorithms seek to discover *statistically significant patterns* (i.e. those with considerable support). They argue, however, that in many investigative

services, including law-enforcement, intelligence and counterterrorism, we may need to find patterns that have no large support but are potentially interesting for a human analyst. The work done by Lin and Chalupski [2003] has a similar motivation. The problem with such attempts is that without statistical significance of the rules the research moves to a more heuristic arena where it is difficult to justify a method and its alerts.

IMBALANCED PATTERNS

The methods suggested in the literature for discovering imbalanced patterns (e.g., in the area of credit card fraud) include both supervised and unsupervised learning. For supervised learning approaches include [Bolton, Hand [2002]:

- minimizing an appropriate cost-weighted loss, and/or
- fixing some parameter (e.g., the number of cases one can afford to investigate in detail) and then trying to maximize the number of fraudulent cases detected subject to the constraints.

If training data with known prior classification of cases as legitimate or fraudulent cases are not available then traditionally unsupervised methods are used. These methods combine profiling and outlier detection. The steps include: (i) modeling a baseline distribution that represents normal behavior and (ii) attempting to detect observations that show the greatest departure from this norm. Digit analysis using Benford's law is an example of such a method. Benford's law (Hill, 1995) says that the distribution of the first significant digits of numbers drawn from a wide variety of random distributions will have (asymptotically) a certain form. Nigrini and Mittermaier (1997) and Nigrini (1999) showed that Benford's law can be used to detect fraud in accounting data. The premise behind fraud detection using tools such as Benford's law is that fabricating data which conform to Benford's law is difficult [Bolton, Hand [2002].

There are important situations where both traditional supervised and unsupervised methods are not appropriate: there is an insufficient set of fraudulent data for supervised learning and there is no data to model a baseline distribution that represents normal behavior for unsupervised learning, because there is no concept of normal behavior in such situations. For instance, Benford's law is not applicable to numbers that are built artificially such as SSN.

Note that *outliers* caused by accidental errors are a rather different from deliberately falsified data [Bolton, Hand, 2002]. This limits applicability of outlier approach in fraud detection. It

can be "... regarded as alerting us to the fact that an observation is anomalous or more likely to be fraudulent than others, so that it can then be investigated in more detail."

Bankruptcy fraud such as purchases using credit cards without intention of paying and leaving the bank to cover the losses is a fraud that reached billions of dollars [Ghosh, Reilly; 1994]. It is one example of the problem of discovering imbalanced patterns because we may have only few training cases where it is proven that bankruptcy was an intentional bankruptcy fraud. The combination of multiple classification rules, and the use of each rule for a suitable environment, were extensively studied in discovering credit card fraud [Stolfo et al, 1997-1999; Wheeler, Aitken, 2000]. Such decomposition can help in solving imbalanced problems if the associated subproblems are balanced.

Provost [2003] reviewed approaches to mitigate imbalanced patterns using traditional data mining technologies. These approaches include: (1) assigning different misclassification costs for false-positive and false-negative errors [(Turney, 2000)], (2) assigning different misclassification error costs for individual cases not only positive and negative categories [Zadrozny, Elkan, 2001], (3) selecting specific portions of positives and negatives for training for training. One of the main difficulties for implementing this approach is that target costs and class distributions may not be known [Provost, 2003].

FUTURE WORK

In our approach we discover and use the most specific rule that is highly probable and statistically significant. The problem is that the quality of such a *best rule* depends on the dataset used to build it. Typically, we do not control the dataset generation, and thus we need to assume that data are *representative* and have no systematic errors. Otherwise, we discover systematic error and we need to analyze discovered rules to *find rules that carry systematic errors*. This is a new research area that is a subject of future study.

An additional future research issue is that some discovered rules/patterns with probability above 0.9 could be abnormal. For instance, the rule may indicate that a Manufacturer sold at a loss a product to a trading company, then the trading company sold it further. Another rule may indicate that a Manufacturer M sold at a loss something to company C that sold its own product to a trading company T. It is not clear from this pattern if manufacturer M sold its product or a precursor. Why would selling at a loss could be a normal pattern? Algorithms and tools need to be developed that would distinguish between highly probable normal patterns and abnormal patterns.

The simulation approach to deal with this problem is to generate another database without suspicious cases, but with negated patterns MBP ⇒ ¬ SR and CBP ⇒ ¬ SR that do not contain suspicious cases. For instance, the case can be: MBP ⇒ BP, a manufacturer bought precursors and then bough more precursors. The difference in probabilities for the rule MBP ⇒ ¬ SR in two databases will show truly suspicious cases.

CONCLUSION

The Hybrid Evidence Correlation (HEC) approach has been outlined in this paper. This data mining technique advances statistical techniques to deal with complex evidence that involve structured objects, text and data of a variety of complex data types that can be numeric and non-numeric. The paper shows potential application of HEC technique for forensic accounting. The technique combines first-order logic (FOL), probabilistic semantic inference (PSI) and negated rules for designing HEC. The approach is illustrated with an example of discovery of suspicious patterns in forensic accounting using simulated data.

The algorithm for finding suspicious patterns based on the main hypothesis (MH) consists of four generalized steps: (1) discovering patterns in the form of probabilistic relational if-then rules in first order logic, (2) negating patterns (then-parts of the rules) and computing probability of each negated pattern, (3) finding records in a database that satisfy negated patterns and analyzing these records for possible false alarms, and (4) removing false alarm records and undertaking detailed analysis of suspicious records.

REFERENCES

1. Agrawal, R., Imielinski, T., Swami A.: "Mining Associations between Sets of Items in Massive Databases", *Proc. of the ACM-SIGMOD 1993 Int'l Conference on Management of Data*, Washington D.C., May 1993, 207-216. http://www.almaden.ibm.com/cs/people/ragrawal/papers/sigmod93.ps
2. Albrecht, W.S. Fraud Examination, Thomson Southwestern, 2003, pp. 145-46.
3. Bolton R., Hand, D., Statistical Fraud Detection: A Review Source: Statist. Sci. 17, iss. 3, 2002, 235–255.
4. Badia, A., Kantardzic, M., Link Analysis Tools for Intelligence and Counterterrorism, In: Intelligence and Security Informatics: Proceedings of IEEE International Conference on Intelligence and Security Informatics, ISI 2005, Atlanta, GA, USA, May 19-20, 2005.
5. Brause,R., Langsdorf, T., Hepp, M., Neural Data Mining for Credit Card Fraud Detection, The Eleventh IEEE International Conference on Tools with Artificial Intelligence Chicago IL, 1999 http://www.informatik.uni-frankfurt.de/~brause/papers/ICTAI99.pdf

6. Chartier, B., Spillane, T. Money laundering detection with a neural network. In Business Applications of Neural Networks (P.J.G. Lisboa, A. Vellido and B. Edisbury, eds.) 159–172. World Scientific, Singapore, 2000,
7. Dzeroski S., Inductive Logic Programming and Knowledge Discovery in Databases. In: Advances in Knowledge Discovery and Data Mining, Eds. U. Fayad, G., Piatetsky-Shapiro, P. Smyth, R. Uthurusamy. AAAI Press, The MIT Press, 1996, 117-152.
8. Durtschi, C., Hillison, W., Pacini, C., The effective use of Benford's Law to assist in detecting fraud in accounting data, *J of Forensic Accounting*, v. V. 2004, 17-34.
9. FSO: Forensic Services: overview, 2002, Ernst and Young LLP, UK, http://www.ey.com/GLOB-AL/gcr.nsf/UK/Forensic_Services_-_overview
10. Getoor, L., Friedman, N., Koller, D., and Pfeffer, A., Learning Probabilistic Relational Models. In Saso Dzeroski and Nada Lavrac, editors. *Relational Data Mining*, Springer-Verlag, New York, New York, 2001.
11. Getoor, L., Link Mining: A New Data Mining Challenge. SIGKDD Explorations, volume 5, issue 1, 2003.
12. Ghosh, S., Reilly, D., Credit Card Fraud Detection with a Neural-Network, 27th Annual Hawaii International Conference on System Sciences (HICSS-27), 1994, pp. 621-630.
13. Grove H., Cook, T. Lessons for auditors: quantitative and qualitative red flags, J. of Forensic Accounting, v. V. 2004, 131-146.
14. Hassibi, K. Detecting payment card fraud with neural networks. In Business Applications of Neural Networks (P. J. G. Lisboa, A. Vellido and B. Edisbury, eds.), 2000, World Scientific, Singapore.
15. Flach, P., Lachiche, N., Confirmation-Guided Discovery of First-Order Rules with Tertius, Machine Learning, 42, 61–95, 2001 http://www.compsci.bristol.ac.uk/~flach/papers/flach-lachiche-mlj01.pdf
16. Fawcett, T., Provost, F., Fraud detection. In Handbook of Knowledge Discovery and Data Mining (W. Kloesgen and J. Zytkow, eds.), 2002, Oxford Univ. Press.
17. Fawcett, T. and F. Provost, "Adaptive Fraud Detection." *Journal of Data Mining and Knowledge Discovery* 1, (3), 1997, http://www.purl.org/NET/tfawcett/papers/DMKD-97.ps.gz
18. Fawcett, T. and F. Provost, F. Activity Monitoring: Noticing Interesting Changes in Behavior. In *Proc. of the Fifth International Conference on Knowledge Discovery and Data Mining* (KDD-99), 1999
19. Forensic Accounting, http://www.in.kpmg.com/services/services_assurance_nav3.html, 2002.
20. Jensen, D. Prospective assessment of AI technologies for fraud detection: A case study. In *Proceedings of AAAI-97 Workshop on AI Approaches to Fraud Detection & Risk Management*, pp. 34–38. AAAI Press., 1997, http://www-eksl.cs.umass.edu /papers/aaaiws97a.pdf
21. IRS forensic accounting by TPI, 2002, http://www.tpirsrelief.com/forensic_accounting.htm
22. Chabrow, E. Tracking The Terrorists, Information week, Jan. 14, 2002, http://www.tpirsrelief.com/ forensic_accounting.htm
23. How Forensic Accountants Support Fraud Litigation, 2002, http://www.fraudinformation.com/ forensic_accountants.htm

24. i2 Applications - Fraud Investigation Techniques, http://www.i2.co.uk/applications/fraud.html
25. Evett, I., Jackson, G. Lambert, JA , McCrossan, S. The impact of the principles of evidence interpretation on the structure and content of statements. *Science & Justice* 2000; 40: 233–239
26. Kantardzic, M., Badia, A., Efficient Implementation of Strong Negative Association Rules. In: A. Wani, K. Cios, K. Hafeez (Eds.): *Proceedings of the 2003 International Conference on Machine Learning and Applications* - ICMLA 2003, June 23-24, 2003, Los Angeles, California, USA, CSREA Press 2003, pp. 152-158.
27. Kovalerchuk, B., Vityaev, E., Data Mining in Finance: Advances in Relational and Hybrid Methods, Kluwer, 2000
28. Kovalerchuk, B., Vityaev, E., Data Mining for Financial Applications, In: Oded Maimon, Lior Rokach (Eds.): The Data Mining and Knowledge Discovery Handbook. Springer 2005.
29. Kovalerchuk, B., Vityaev E., Ruiz J.F., Consistent and Complete Data and "Expert" Mining in Medicine, In: Medical Data Mining and Knowledge Discovery, Springer, 2001, pp. 238-280.
30. Kovalerchuk, B., Vityaev E., Ruiz J.F., Consistent Knowledge Discovery in Medical Diagnosis, *IEEE Engineering in Medicine and Biology Magazine*, (Special issue on Data Mining and Knowledge Discovery), vol. 19, N, 4, July/August 2000, pp. 26-37.
31. Krantz, D.H., Luce, R.D., Suppes, P., Tversky, A. (1971, 1989, 1990), Foundations of measurement, Vol. 1,2,3 - NY, London: Acad. press, (1971) 577 p., (1989) 493 p., (1990) 356 p.
32. Last, M., Kandel, A., Automated Perceptions in Data Mining, 1999 IEEE International Fuzzy Systems Conference Proceedings, Part I, pp. 190 - 197, Seoul, Korea, August 1999. http://citeseer.ifi.unizh.ch/cache/papers/cs/10335/http:zSzzSzwww.csee.usf.eduzSz%7EmlastzSzpaperszSzperc_f1.pdf/last99automated.pdf
33. Lin, S., Chalupsky, H., Unsupervised Link Discovery in Multi-relational Data via Rarity Analysis, In: Proc. The Third IEEE International Conference on Data Mining, ICDM '03, Melbourne, Florida, USA, November 19 - 22, 2003.
34. Mena, J. Investigative Data Mining for Security and Criminal Detection, Butterworth-Heinemann, 2003.
35. Mitchell, T., Machine Learning, McGraw Hill, 1997
36. Muggleton, S., Learning structure and parameters of stochastic logic programs. *Electronic Transactions in Artificial Intelligence*, 6, Vol. 7, nr 016, 2002
37. Neville, J., Jensen, D., Friedland, L., and Hay, M. Learning relational probability trees. In *Proc. of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.625–630, 2003.
38. Prentice, M., Forensic Services - tracking terrorist networks,2002, Ernst & Young LLP, UK, http://www.ey.com/global/gcr.nsf/UK/Forensic_Services_-_tracking_terrorist_networks
39. Pazzani, M., Knowledge discovery from data, *IEEE Intelligent Systems*, 15(2): 10-13, 2000.
40. Provost, F., The Role of Applications in the Science of Machine Learning, 2003, ICML-2003, Washington, DC, http://pages.stern.nyu.edu/%7Efprovost/Papers/ICML-2003-distr.pdf
41. Provost, F., Domingos, P., Tree induction for probability-based rankings. Machine Learning, 52:3, 2003.

42. Provost, F. Learning with Imbalanced Data Sets 101, AAAI'2000 Workshop on Imbalanced Data Sets, 2000, http://pages.stern.nyu.edu/%7Efprovost/Papers/skew.PDF
43. Prentice, M., Forensic Services - tracking terrorist networks,2002, Ernst & Young LLP, UK, http://www.ey.com/global/gcr.nsf/UK/Forensic_Services_-_tracking_terrorist_networks
44. Puech, A., Muggleton, S., A comparison of stochastic logic programs and Bayesian logic programs. In IJCAI03 Workshop on Learning Statistical Models from Relational Data. IJCAI, 2003.
45. Rattigan, M., Jensen, D., The Case for Anomalous Link Detection, In the Proceedings of the Fourth International Workshop on Multi-Relational Data Mining (MRDM-2005), Aug. 21, 2005, Chicago. http://kdl.cs.umass.edu/papers/rattigan-jensen-mrdm2005.pdf
46. Rosenthal, H., Fraud and the Auditor In the Real World, Forensic Accounting Information and Education Center, LLC, 2001, http://www.askhal.com/fraud.html
47. Samokhvalov, K., On theory of empirical prediction, Computational Systems, #55, 1973, pp. 3-35 (In Russian)
48. Thuraisingham, B., Web Data Mining and Applications in Business Intelligence and Counter-Terrorism, CRC, 2003.
49. Turney, P., Types of cost in inductive concept learning. In *Proceedings Workshop on Cost-Sensitive Learning at the Seventeenth International Conference on Machine Learning* (WCSL at ICML-2000), 2000, pp.15-21.
50. Vangel, D., James, A., Terrorist Financing: Cleaning Up a Dirty Business, the issue of Ernst & Young's financial services quarterly, Spring 2002. http://www.ey.com/GLOBAL/content.nsf/International/Issues_&_Perspectives_-_Library_-Terrorist_Financing_Cleaning_Up_a_Dirty_Business
51. Vityaev E.E. Semantic approach to knowledge base creating. Semantic probabilistic inference of the best for prediction PROLOG programs by a probability model of data. In: *Logic and Semantic Programming* (Computational Systems, v.146), Novosibirsk, 1992, p.19 49. (in Russian)
52. Weatherford. M., Mining for Fraud, *IEEE Intelligent systems*, Vol. 3, N 7, July-Aug., 2002
53. Weiss, G., Mining with rarity: a unifying framework, June 2004 *ACM SIGKDD Explorations Newsletter*, Vol. 6, Issue 1.
54. Will, H.J. ACL: a language specific for auditors, *Communications of the ACM*, Vol. 26, Issue 5, 1983, pp. 356 – 361.
55. Zadrozny, B., Elkan, C., Learning and Making Decisions When Costs and Probabilities are Both Unknown. In *Proc. of the Seventh Intern.Conference on Knowledge Discovery and Data Mining* (KDD'01), 2001
56. Zagoruiko N.G., Elkina V.N. Eds., Machine methods for discovery regularities, *Proceedings of the MOZ'76 Conf.*, 1976, Novosibirsk (In Russian)