IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Suite 200, Hershey PA 17033, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITP5251

This paper appears in *Managing Modern Organizations Through Information Technology*, Proceedings of the 2005 Information Resources Management Association International Conference, edited by Mehdi Khosrow-Pour. Copyright 2005, Idea Group Inc.

Detecting Gains from Service Criminal Activities Using Generic Algorithms for Money Laundering Detection

Christoph Blodig, Tino Jahnke and Juergen Seitz

Dept of Elect. Commerce/Elect. Business, Mobile Business/Telematics, University of Cooperative Education Heidenheim, Schmelzofenvorstadt 33, 80520 Heidenheim, Germany, seitz@ba-heidenheim.de

ABSTRACT

This paper covers the field of money laundering detection in Germany. The demand for the approach has been initially developed by the BMW Cash Group, Munich where severe criminal activities like money laundering need to be monitored automatically. Therefore, a statistical rule-based approach is discussed in the first part of the paper. In order to enhance the monitoring concept a genetic algorithms approach is applied to design a real-time monitoring framework.

INTRODUCTION AND MOTIVATION

Governments take organized criminal activities and terrorism more and more seriously. As a reaction to the events of September 11th, the Financial Action Task Force (FATF) declared an expansion of its mission (FATF, 2004a). While organized criminal activities are directly dedicated to earn massive benefits, terrorism is focused on other primary targets. But therefore, they need also voluminous funds (Kilchling, 2004; Fijnaut, Paoli, 2004; Kilchling 2002). This increases the pressure on financial service providers to participate actively in avoiding organized criminal activities and terrorism by detecting benefits from serious criminal activities. This is on the one hand side tried through expanding disclosure responsibilities. On the other hand financial service providers receive duties to perform monitoring of suspicious transactions. In 1993, Germany created a law covering the detection of profits from serious criminal activities, called “Geldwaeschegesetz” (GWG) (German Money Laundering Act). This law requires credit institutions, financial service providers as well as lawyers, auditors, brokers and casinos to identify all persons involved in transactions above 15.000 •. This also applies to multiple transactions, which together exceed the amount within a short period of time.

According to § 14 GWG, credit institutions, insurance companies, auctioneers, financial service providers, investment share companies, noble metal dealers and casinos have to have precautions in place not to be abused for money laundering. This includes naming a money laundering commissioner, the development of internal principles, security and control systems and the development of other organizational measures. These obligations also apply to branch offices and independent companies according to § 18 of the German Stock Companies Act and show wide disclosure duties. In addition to suspicious transactions with incoming payments and payouts of 15,000 •, there are soft rules in place. In a number of comments not only the specific value of 15,000 • and above is to disclosure. Furthermore, there have been established numerous generally accepted inspection rules out of the soft defined abuse prevention duties. Many of the obligations are transferred recommendations of the FATF (FATF 2003).

There are a number of legal and reputation-related risks for financial service providers resulting from the Money Laundering Act. According to given law, they are as information holders liable for taking preventive, corrective and reporting actions. The use of applicable methods and tools is explicit desired.

Figure 1 shows possible starting points for a money laundering (ml) examination.

After a customer performed an action, the involved employee can initialize a notification process if there is any suspicion. Anyway, a system based check-up for the action takes place. Only on that the action is processed. The underlying information is checked again in analytical system-based inspections. System-risen suspicions must be checked by the responsible employee if an statistical alpha or beta error occurred, i. e. the employee noticed a suspicious transactions and the system didn’t, or the system alerts a money laundering suspicion which the employee cannot verify. If the analysis results in an ongoing suspicion, the notification process is initialized. Due to duties of the Money Laundering Act to delay or avoid transactions with a severe money laundering suspicion, first internal control and security mechanisms are integrated already in the transaction systems. But the main focus of this paper targets the second system-based area of internal security mechanisms, where the ex-post check-up of the entire database takes place. They use all transactions. Current check-ups use a predefined set of different rules, which are designed to recognize money laundering suspect behavior in the actions of an account or an account owner. This rule set is mostly static and is used by a lot of institutions. The fact that the opposite party shows intelligent behavior is at this point no more trivial for the aspects of success and effectiveness.

Figure 2 shows an example of a widely used rule-base.

These rules are an example set (BMW Financial Services Germany, 2004). They may fairly easy be implemented and can be applied to existing transaction data using simple SQL queries for example. Scoring values are added to the according account owner on the list of returned account owners by the SQL queries. Account owners with high scoring values can be easily and effectively identified then. Well-informed opponents can avoid these recognition methods. Depending on the rule, the degree of difficulty can be very different. Leaving the purpose of a

Figure 1. Money Laundering Examination Process

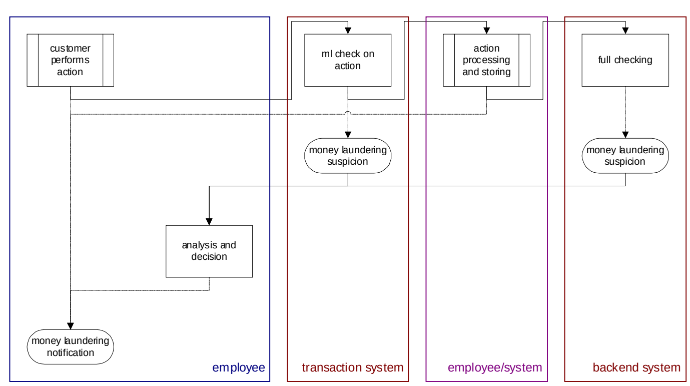

Figure 2. Generic Detection Rules Examples

Items to be checked	SCORE
Nationality of account owner in NCCT list	2
New customer with high cash deposits	2
Name of account owner in VIP list (including alias)	2
High number of cash deposits	1
High cash deposits	1
High number of payouts	1
High payouts	1
Nationality of account owner on watch list (not NCCT)	1
Many negative transactions	1
High negative transaction volume	1
Collective accounts	2
High ratio of cash deposits (>15.000)	2
Smurfing (high sum of cash deposits within 9 days)	2
Many transactions towards dedicated receivers	1
One high cash deposit within 30 days	1
Unknown/empty subject	1
Name of sender/receiver in VIP list	2
Nationality of sender/receiver in NCCT list	2
Sudden rise of transaction volume	1
High amount of different cash depositors	1
More than five transactions with empty/unknown applicant	1
High anticipation amount	2
Cash anticipation, high amount in foreign currency	2
High amount of a load, short duration	2
Short duration of a current account	1
High amount of visits in different branch offices	1
Collective account cash deposits	2

transaction empty is easily avoided, but to evade smurfing detection is much more difficult if there is money laundering to do. Altogether, money laundering is made more difficult through the use of analytical processing, but well-informed groups are still able to exhaust the known rules and scoring methods depending on their risk affinity. They have a broad set of strategic possibilities between the consequent evasion of all known rules or the calculated pass-the-tolerance-value way. This classic method is quite very successful, because it is easy to apply and money laundering can be heavily reduced by it. Anyhow, there is a need for another detection method, which is more dynamic. There are two main requirements that are not yet covered by the classic method. These are learnability and unpredictability. Learnability should enable the algorithm to react on evasion trials and to implement them in the recognition process. This should eliminate the weakness of static detection logic. Whereas the aim of unpredictability is to ensure in the sense of present intelligent opponents, it must not be possible to predict whether detection is triggered or not. This should eliminate the weakness of determinable detection logic. In order to design and create a working prototype, the concept of genetic algorithms can be applied.

GENETIC ALGORITHMS

Genetic algorithms base upon the basics of evolutionary theory und belong to stochastic search methods (Pohlheim, 1997; Laemmel & Cleve, 2001). Stochastic search methods try to approximate to the optimal solution step-by-step, starting from one or more initial solutions. Unlike the technical implementation, the underlying concept is rather simple. Genetic algorithms are geared to the evolutionary methods of nature, which created various individuals and suited them to their environment along generations (Sipper, 2002; Michalewicz, 1996; Mitchell, 1996). The results are perfectly to their habitat adjusted life forms. Humans seem to be the most perfect result of evolution. Eyes for example are perfect arranged, structured and protected. They can perceive 600 million colors and eclipse every high-tech light-measuring unit. Genetic algorithms have largely been developed by John Holland and fellows and students of the University of Michigan (Holland, 1975). By implementing genetic algorithms in software, generations of software individuals continuously adapt to the aim (Banzhaf et al., 1997; Mitchell, 1996). The general approach will be expatiated in the following chapters.

General Genetic Functions

Genetic algorithms are stamped by simpleness and success (Pohlheim, 1997). For the transformation of a simple genetic algorithm there is nearly nothing more needed than strings, which are copied, split and assembled. Below there will be taken up a popular example, which can simulate a population of strings (Laemmel & Cleve; 2001; Pohlheim, 1997; Bigus & Bigus, 2001). The strings in figure 3 represent binary encoded individuals of a population in their initial stage and are created at random. In this regard the term chromosome is often used, which has exactly this meaning in the following. To go on with creating a successful population, there is an effective genetic algorithm consisting of a fitness function and a number of functions for crossover, reproduction, selection and mutation (Pohlheim, 1997; Goldberg, 1989; Laemmel & Cleve; 2001). The fitness function is a function that represents success, goodness, quality and usability of the results respectively the configuration. The term fitness was stamped by Darwin and relates in this context to the survivability of individuals (Goldberg, 1989). The fitness function plays an important role in the production of new generations, because the selective function is mostly based on it.

Selection

Selection describes the results of the fitness function and the dominance of “good” individuals for being transferred into the new generation (Laemmel & Cleve, 2001). The selection of suitable parents for the creation of new generations is usually determined by various scientific methods. Often described procedures include the “roulette wheel” and competition technique (Pohlheim, 1997; Goldberg, 1989; Obitko 1998). Based on the approach that a generation is capable to generate a lot of good individuals, there must be a selection that is advantageous for them and harmful for the others (Miller & Goldberg, 1996). Whereas “roulette wheel” selects at random after adjusting the probabilities by fitness, the competition approach enhances the best individuals of random tournament groups.

Reproduction

Reproduction describes the process of the creation of new individuals for a continued evolution of individuals. It is also closely tied to the properties of selection, especially its selection procedures and is often described as selection.

Crossover

The purpose of crossover is the creation of better solutions. For this purpose good solutions are being combined with each other to create better solutions. There are multiple approaches for crossovers. Known crossover variations are single point and double point crossovers. The procedures are shown in Figure 3.

Mutation

The mutation is used to try to find totally different solutions than those that may be found by using crossover like mentioned above. So one aspect of mutation is to ensure the variety in the population. Furthermore, mutation is used to avoid the creation of populations that are

Figure 3. Single Point and Double Point Crossover

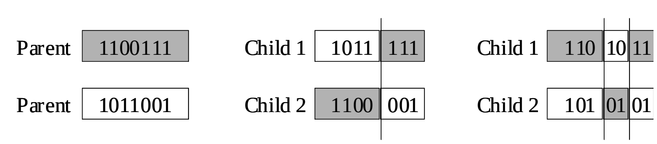

Figure 4. Genetic Algorithm (Goldberg, 1989)

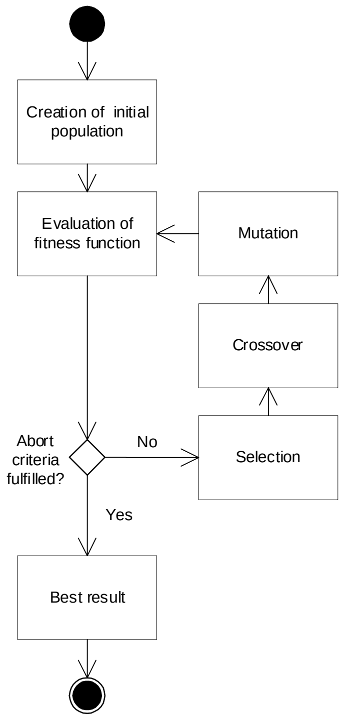

Figure 5. Pseudo-Code of the Genetic Algorithm

```
Algorithm GA is

  // start with an initial time
     t := 0;

  // initialize a usually random population of individuals
     initpopulation P (t);

  // evaluate fitness of all initial individuals of population
     evaluate P (t);

  // test for termination criterion (time, fitness, etc.)
     while not fitEnoughForTermination P (t) do

       // increase the time counter
        t := t + 1;

       // select a sub-population for offspring production
        P' := selectparents P (t);

       // recombine the "genes" of selected parents
        recombine P' (t);

       // perturb the mated population stochastically
        mutate P' (t);

       // evaluate its new fitness
        evaluate P' (t);

       // select the survivors from actual fitness
        P := survive P,P' (t);
  od
end GA
```

unable to evolve, for example through exponential rising clone effects with identical parents. Mutation evolves randomly. Whereas crossover favors a more local search, mutation provides a way of performing a mixed-in global search (Thierens & Goldberg, 1994; Goldberg, Korb, & Deb, 1989).

Genetic Algorithm

General Algorithm
Figure 4 shows a general genetic algorithm.

The genetic algorithm shown in figure 4 operates as follows:

- Create a random population consisting of individuals that represent a potential solution of the problem.
- Calculate the fitness f(x) of each individual. This means the quality of each solution.
- New generation (Selection): Choose the two individuals out of the population that have the highest fitness.
- Crossover: Recombine the parents with to a crossover probability, so that new offspring is created. If a crossover is not being executed, the child is an exact copy of its parent.
- Mutation: Change the individuals in each position using a mutation probability.
- Exchange the old generation with the new generation. Regard the new generation as the starting basis. Calculate the fitness function as shown in the second step. Continue at step two.
- When the abort condition is fulfilled and an acceptable solution has been found, stop the algorithm and show the solution.

Figure 5 shows the pseudo-code of the algorithm.

Applications of Genetic Algorithms
Genetic algorithms can be used in a variety of applications in very different areas. They are used in particular for optimization problem (Pohlheim, 1997). Generally, genetic algorithms are used for problems that search for acceptable, not perfect solutions. Even Microsoft optimizes its program code in parts using genetic algorithms. Genetic algorithms are used especially for pattern recognition, adaptive game programming, biological simulation methods and optimized search methods (Pohlheim, 1997; Goldberg, 1989; Obitko, 1998).

Genetic Algorithms in Learning Agents
Intelligent software agents, who cover there learning processes with genetic algorithms, begin with a population of random assembled strategies and use stand-alone or cooperative matching. During this process, results are rated and the best results turn into the base material for the development of new strategies (Beam & Segev, 1996). Therein is a high potential for gaining suitable solutions based on effective independent strategies. The lack of known faultless detection methods results in a tolerance covering the deformities of genetic algorithms. While the import aspect is that we need a solution to detect suspicious cases, and don't need to know exactly how the solution is found, the application of genetic algorithms to solute seeking problems is acceptable for this kind of purpose (van Hemert, 2001).

GENETIC ALGORITHMS FOR MONEY LAUNDERING DETECTION

The named detection criteria for money laundering to seek suspicious activities are used in the form of expressions. These expressions contain a parameter set. The predefined setting of these parameters is responsible for the static and determinable behavior of the classic detection method. The check-up for "Nationality of account owner in Non Cooperative Countries and Territories (NCCT) list" is implemented as a simple in-list check against a pre-set array of all yearly named nations (FATF 2004b) according to the FATF criteria (FATF 2000) with a Boolean value, where the NCCT countries of them are set to true. All check expressions consist of a constant constraint and one or more changeable parameters that range in a given plausible domain.

Below two examples for the money laundering examination are shown in relational algebra. The checking expression returns the customers id which then earn the additional score for this term.

Box A.

$$\pi_{tcid}(\sigma_{cash>total \cdot 0.25}($$
$$\gamma_{ccid=cid,cash=COUNT(tid)}(Customers \times_{cid=acid} Accounts \times_{aid=taid} \sigma_{ttype='cash' \cap tamount>0}(Transactions))$$
$$\times_{ccid=tcid} \gamma_{tcid=cid,total=COUNT(tid)}(Customers \times_{cid=acid} Accounts \times_{aid=taid} Transactions)))$$

Box B.

$$\pi_{cid}(\sigma_{emptysubjets>=5}($$
$$\gamma_{cid,emptysubjets=COUNT(tid)}(\sigma_{cid=CTC}(Customers) \times_{cid=acid} Accounts \times_{aid=taid} \sigma_{tsubject=''}(Transactions)))$$

Figure 6. Basic Individuals

Individual rule	**Parameter**	**Example value**
High ratio of cash deposits	Cash ratio	0.25
Many empty subjects	Empty-subjects	5.00

For these examples the following relations are used:

Customers (cid, cname, cgivenname)

Accounts (aid, acid)

Transactions (tid, taid, tamount, tfromto, tsubject)

Check on high ratio of cash deposits (see Box A).

Check on empty subjects (see Box B).

This leads to two basic individuals (Figure 6).

By applying genetic algorithms, these parameters may be optimized. The population of these checking rules will be called "Checkers". Crossovers only take place between "Checkers" based on the same underlying check expression.

To achieve an acceptable substitute for the classic scoring function, a second population is created. This leads to the rating „Reporters". Their chromosomes are two references on checkers, and they identify suspicious cases only, if both adapted "Checkers" insert a signal. It is also possible that both references link towards the same "Checker".

Mutation is the random inclusion of a new „Checker". For enabling them to take part early in the rating process, mutation should occur very often in the "Reporters" population. It's also worth to consider the possibility of linking another "Reporter" instead of a "Checker".

There is a huge amount of historical data, so the population can perform a lot of development. The rating is based on competitive success measurement. Valid suspicions, which have been only made through the genetic detection method, should receive special rewards.

The fitness function first selects bad "Reporters" by the use of statistic methods. Then the selection function eliminates "Reporters" by the regard of both, random and fitness value. The ratio of killed connected "Reporters" will turn into the fitness value of "Checkers". But due to the different races of "Checkers" there should be established an additional minority protection.

As a consequence, the genetic detection has the possibility to develop better parameter values, arrange a weight-based cooperation of the different checking clauses and enhance correlating individuals. This results in a higher ratio of money laundering detection, and will also weaken detection errors. Furthermore, as the method is used also in combination with examinations by humans and over time, learning and adapting is a big advantage. Since there are a certain number of spot tests, even a few runaways are welcome.

CONCLUSION

Genetic algorithms offer an excellent way of detecting and hunting money laundering activities. They should be used as described in addition to the classic rule-based detection to extend the pre-selection with two important factors. On the one hand, there won't be the problem of static detection criteria any longer. A kind of learning behavior is added to the main issue and allows quick adaptation to new ways of hiding gains from illegal business. On the other hand, the transparency of the classic methods is not any more of use to the money launderers. They will have difficulties to create transactions sets that will certainly not cause a further check by evading pre-selection criteria. The additional use of genetic money laundering detection will make life for organized crime and terrorism much harder. At least, the approach needs to be tested with a prototype application and either a batch- or real-time monitoring system have to be developed.

REFERENCES

Banzhaf, W., Nordin, P., Keller, R. E., & Francone, F. D. (1997). Genetic Programming - An Introduction: On the Automatic Evolution of Computer Programs and its Applications. San Francisco: Morgan Kaufmann Publishers.

Beam, C., & Segev, A. (1996). Electronic Catalogs and Negotiations, CTIM Working Paper 96-WP-1016. Berkley: The Fisher Center for Management and Information Technology.

Bigus, J., & Bigus, J. (2001). Intelligente Agenten mit Java programmieren-eCommerce und Informationsrecherche automatisieren. Munich: Addison-Wesley.

BMW Financial Services Germany (2004). Internal report on money laundering requirements and technologies. Munich.

FATF (Financial Action Task Force on Money Laundering) (2000). Report on Non-Cooperative Countries or Territories. Received January 4th, 2005 from http://www1.oecd.org/fatf/pdf/NCCT_en.pdf.

FATF (Financial Action Task Force on Money Laundering) (2003). The forty recommendations of the Financial Action Task Force on Money Laundering. Received January 4th, 2005 from http://www.oecd.org/fatf/pdf/40recs-2003_en.pdf.

FATF (Financial Action Task Force on Money Laundering) (2004a). Received January 4th, 2005 from http://www.oecd.org/fatf/.

FATF (Financial Action Task Force on Money Laundering) (2004b). Annual Review of Non-Cooperative Countries or Territories. Received January 4th, 2005 from http://www1.oecd.org/fatf/pdf/NCCT2004_en.PDF.

Fijnaut, C., & Paoli, L. (eds.) (2004). Organised Crime in Europe: First Systematic Comparison: Concepts, Patterns and Control Policies in the European Union and Beyond. Dordrecht: Springer.

Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. New York: Addison-Wesley.

Goldberg, D. E., Korb, B., & Deb, K. (1989). Messy genetic algorithms: motivation, analysis, and first results. In: Complex Systems, vol. 3, no. 5, pp. 493 – 530.

Hemert, J. I. van (2001). Evolutionary Computation in Constraint Satisfaction and Machine Learning. Leiden Institute of Advanced Computer Science, Leiden University.

Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. Ann Arbor, Michigan: The University of Michigan Press.

Kilchling, M. (2002). Die Praxis der Gewinnabschoepfung in Europa. Eine vergleichende Evaluationsstudie zur Gewinnabschoepfung in Faellen von Geldwaesche und anderen Formen Organisierter Kriminalitaet. Freiburg i. Br.: Max-Planck-Institut fuer auslaendisches und internationales Strafrecht.

Kilchling, M. (2004). Die Bedeutung der Geldwaeschebekaempfung als gezielte Strategie zur Bekaempfung der Organisierten Kriminalitaet in der aktuellen Diskussion: Theorie und Praxis. In: Gehl, G. et al. (ed.). Geldwaeschebekaempfung, Zeugenschutz, Gewinnabschoepfung – Wege zur Bekaempfung der Organisierter Kriminalitaet? Ein europaeischer Vergleich. Weimar: Bertuch-Verlag, pp. 27 - 44.

Laemmel, U., & Cleve, J. (2001). Lehr- und Uebungsbuch der Kuenstlichen Intelligenz. Leipzig, Munich: Fachbuchverlag.

Michalewicz, Z. (1996). Genetic Algorithms + Data Structures = Evolution Programs. Berlin: Springer.

Miller, B. L., & Goldberg, D. E. (1996). Genetic Algorithms, Selection Schemes, and the Varying Effects of Noise. In: Evolutionary Computation, vol. 4 no. 2, pp. 113 - 131. Received January 5th, 2005 from http://citeseer.ist.psu.edu/article/miller96genetic.html.

Mitchell, M. (1996). An Introduction to Genetic Algorithms. Cambridge, MA: MIT Press.

Obitko, M. (1998). Introduction to Genetic Algorithms. Received October 3rd, 2004 from http://cs.felk.cvut.cz/~xobitko/ga/.

Pohlheim, H. (1997). Evolutionary Algorithms: Principles, Methods and Algorithms. Technical Report, Technical University Ilmenau, 1994 – 1997.

Sipper, M. (2002). *Machine Nature: The Coming Age of Bio-Inspired Computing.* New York: McGraw-Hill.

Thierens, D., & Goldberg, D. E. (1994). Convergence models of genetic algorithm selection schemes. In: Davidor, Y., Schwefel, H.-P., & Manner, R. (eds.). Parallel Problem Solving from Nature - PPSN III. Berlin: Springer, pp. 119 – 129.

Related Content