An Approach for Automatic Fraud Detection in the Insurance Domain

Alexander Widder[1], Rainer v. Ammon[2], Gerit Hagemann[3], Dirk Schönfeld[4]

[1] simple fact AG, D-90491 Nuremberg, Germany, alexander.widder@simplefact.de,
[2] Centrum für Informations-Technologie Transfer GmbH, D-93051 Regensburg, Germany, rainer.ammon@citt-online.com,
[3] metafinanz Informationssysteme GmbH, D-80804 Munich, Germany, gerit.hagemann@metafinanz.de,
[4] metafinanz Informationssysteme GmbH, D-80804 Munich, Germany, dirk.schoenfeld@metafinanz.de

Abstract

A new approach to detect fraudulent event patterns in the field of insurance fraud detection by using a combination of discriminant analysis and neural network techniques is presented. The approach is embedded in a Complex Event Processing (CEP) engine. CEP is an emerging technology for detecting known patterns of events and aggregating them as complex events at a higher level of analysis in real-time. In the insurance domain, fraud detection often is a manual task and automatically fraud detection contains an enormous potential for streamlining and saving costs.

Introduction

A large part of the population considers insurance fraud as a trivial offense, expressing an attitude that almost lacks any sense of wrongdoing. There is often little hesitation in claiming a broken pair of glasses through the personal liability insurance policy of a friend, or exaggerating the loss incurred. Organized criminals are also active insurance fraudsters. Especially in the field of third-party liability insurance for automobiles, the sheer mass of claims makes it a relatively simple task to stage fraudulent claims and get paid. There are many types of fraud, and one can classify the acts of deception into the following categories:

- Fabricated accidents: An accident either did not occur, or at least not as stated. One merely asserts that it did occur in order to have a legitimate claim.
- Exploited accidents: Here, an actual accident did occur and is exploited to get reimbursed for preexisting damage, or the damage is intentionally increased to gain some advantages.
- Staged accidents: A collision did take place but, if one strictly applies the laws of coincidence, an accident did not really happen. At times, rental vehicles are used, and often the same vehicle is involved in several incidents. The damage to the vehicle is either not repaired, or only to the extent absolutely necessary.
- Provoked accidents: One driver intentionally involves another innocent driver in an accident – which is crafted cleverly to make the latter appear as the one at fault. For example, a driver accelerates briefly before a yellow light and brakes hard, or perhaps reverses in front of a red light. Blind corners are favorite sites for such accidents, and accomplices are always on hand to coordinate the accident and serve as witnesses.

The *Berlin Model* is one unique manifestation of a staged accident. Dating back to the 80s, this method was frequently applied in Berlin's district of Kreuzberg. A stolen vehicle would be used to damage one or more cars during the night. The stolen vehicle was then abandoned at the site of the accident, to facilitate finding the holder of the third party insurance policy. It is hard to get reliable or concrete figures on the number of fraudulent claims, since insurers are unwilling to admit publicly how easily they were taken for a ride. There are some estimates, however, which suggest that about 10% of all comprehensive insurance claims contain elements of suspected fraud [4, p. 1490].

Detecting Insurance Fraud - The state of the art

All insurers employ specialists to process suspicious claims. The problem, of course, is to recognize a fraudster among the sheer number of claims filed. But one can apply certain criteria to select claims with a higher probability of being fraudulent. For instance:

- The parties involved know each other or are close neighbors
- There are no witnesses, because the site of the accident is isolated and/or the accident occurred in the dark
- A forged/false document is submitted for the damage claim
- The policyholder's vehicle is old and of low value

- The damaged vehicle is old but of a premier class
- The vehicles are no longer available to reconstruct the accident scene
- The cause of the accident is hard to follow
- The policyholders make contradictory statements

Most often, recognizing insurance fraud depends on claims adjusters and their gut feelings. To do some automated analysis, one requires an adequate database with sufficiently structured information. Although most insurers have good computing systems for settling claims, the task at hand is to technically identify the factors that would flag a possibly suspicious claim.

Many attempts have been undertaken in this field, including the application of complex rules and neural networks. In fact, some insurers even use neural networks in their daily operations, but they are understandably secretive about all this. This is probably because they do not want to loose their advantage by revealing too much about this methodology to fraudsters. Fundamentally, a suspected case of fraud comes to light only after the claim has reached a somewhat mature stage. This is the stage after basic details of the claim, the parties involved, the policy concerned, the cause of the accident, and the amount of the damage incurred have been identified and cleared. A claims adjuster, who can supplement the information on the case, may also have been assigned to take a look. This is the standard procedure for all insurance companies. If it turns out that the circumstances require the insurer to issue a check for the claim, the payment will be authorized. However, once the money has been paid, it is hard to recover it. That is why it is critical to do one's homework and check for fraudulent intentions – before reaching this stage.

Detecting fraud attempts by combining discriminant analysis and neural networks

The new approach in fraud detection combines discriminant analysis (see [2]) and neural networks (see [3]). The advantage is that every event represents one input value of a neural network. The CEP engine creates event clusters based on known historical fraud and non-fraud events for specific training customers. The allocation of an event in a specific cluster depends on event attributes relevant for classifying an event as fraud or non-fraud. By using the values of these relevant attributes for calculating the discriminant coefficient, the discriminant function will be computed. The discriminant function is used for allocating a new occurring event into a specific group of events. This is achieved by inserting the relevant attribute values of a new occurring event in the discriminant function and comparing the computed value with the critical discriminant value based on the historic event clusters. The accurate definition of the allocation process can be found in [5]. At the next stage the discriminant values are analyzed by a neural network. The weights of the network are determined by training with discriminant values from known fraud and non-fraud event patterns of specific training customers. So the discriminant values are used as input values for the neural networks. The possible frequency of the training processes depends on the performance of the detection system. If this process is leading to a decrease of the system performance, it can be regulated e.g. by running grid computing techniques [1]. One discriminant value represents one event of a pattern that should be distinguished as fraud or non-fraud by the neural network. After running the neural network for an occurring combination of event discriminant values, the output value will be evaluated in order to divide fraudulent from non-fraudulent combinations. For known fraud combinations, the networks are trained with 1 as output value, whereas known non-fraud combinations are trained with 0. In order to identify unknown combinations, a threshold is determined based on the training results e.g. 0.5. If the output value of an input combination of events (respectively discriminant values) is greater than the threshold, the system classifies it as a suspected fraud and reacts with a predefined action e.g. sending an alert to an operator or moving the case to a specialized group. The architecture is described exactly in [6].

Fraud Detection in the claims settlement of the insurance domain

Regarding the fraud detection model, described in the previous chapter, the first step in identifying insurance fraud attempts is to select the relevant features respectively attributes of the event. The claim message events contain attributes just as:

- The estimated total loss, incident time and loss location
- the personal data of the causer of the loss
- the personal data of other ones involved like claimant and witnesses
- the description of the succession of the incident

These are important attributes for detecting fraud attempts in the claims settlement, but this information is not enough. The events have to be enriched with additional attributes regarding the specific claim. These additional attributes contain information about the policy and the claim history of the insurant as well as the environment of the claim, just as:

- the policy period
- the total of previous claim losses due to the insurant
- weather reports at incident time

In order to enrich the events, the necessary information has to be loaded for example from external databases at runtime. On the one hand, the numerical attributes like “policy period”, “number of previous claims” or “total loss of actual claim” are used for discriminant analysis in the fraud detection component of the CEP engine. On the other hand, the non-numerical attributes like “weather data” or “loss location” etc. are used for the decision tree of the hybrid fraud detection model. The decision tree can be generated automatically on the base of test data respectively training data, whose fraud state is already known. The neural network will be trained with the training data before running the fraud detection system by using the backpropagation algorithm. Backpropagation is a supervised learning algorithm for neural networks and is based on the gradient descent method, see [3]. The network can be trained in a parallel process during runtime. After finishing training, the weights can be updated at runtime. The fraud detection system does not need to be stopped for learning new patterns.

References

[1] Berman, F., Fox, G., and Hey, A. 2003. Grid Computing – Making the Global Infrastructure a Reality. West Sussex: John Wiley and Sons Ltd.

[2] Mardia, K.V., Kent, J. T., and Bibby, J. M. 1979. Multivariate Analysis. San Diego, San Francisco, New York, Boston, London, Sidney, Tokyo: Academic Press.

[3] Rojas, R. 1996. Neural Networks - A systematic Introduction. Berlin, Heidelberg, New York: Springer Verlag.

[4] Ulbricht, N., and Fähnrich, E. 2005. Industrialisierung in der Betrugsbekämpfung – Noch enorme Nettoeinsparpotenziale durch mehr Effizienz und Effektivität von Betrugsabwehrstrategien. Cologne: PMG Presse Monitor Deutschland.

[5] Widder, A., Ammon, R. v., Schaeffer, P., and Wolff, C. 2007. Identification of suspicious, unknown event patterns in an event cloud. In *Proceedings of the inaugural international conference on Distributed event-based systems*, 64 – 70. Toronto: ACM.

[6] Widder, A., Ammon, R. v., Schaeffer, P., and Wolff, C. 2008. Combining Discriminant Analysis and Neural Networks for Fraud Detection on the Base of Complex Event Processing. In *Proceedings of the 2nd international conference on Distributed event-based systems*. Fast abstract paper, Rome: ACM.