Combining Discriminant Analysis and Neural Networks for Fraud Detection on the Base of Complex Event Processing

Alexander Widder[1], Rainer v. Ammon[2], Philippe Schaeffer[3], Christian Wolff[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] TÜV Rheinland Secure iT GmbH, D-51105 Cologne, Germany, philippe.schaeffer@de.tuv.com,
[4] Media Computing, University of Regensburg, D-93040 Regensburg, Germany, christian.wolff@sprachlit.uni-regensburg.de

ABSTRACT

A new approach to detect suspicious, unknown event patterns in the field of 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. Detection systems can be differentiated in rule based systems and those based on statistical methods. In order to reach the goal of finding unknown fraud patterns, several statistical methods are discussed. On this background, first experimental results of our approach as a combination of CEP, discriminant analysis and neural networks are presented.

Categories and Subject Descriptors

I.5.2 [**Pattern Recognition**]: Design Methodology – *Pattern analysis*; H.4.2 [**Information Systems Applications**]: Types of Systems – *Decision support (e.g., MIS)*

General Terms

Algorithms, Design, Experimentation, Security

Keywords

Complex Event Processing, Discriminant Analysis, Neural Networks, Fraud Detection, Event Patterns

1. INTRODUCTION

In the global event cloud of an organization many kinds of events occur. According to [2, 3, p. 88] an event is a record of an activity in a system and may be related to other events. By the use of CEP-engines, low-level events can be aggregated to high level events in real time. This can be achieved with *known* event patterns. Known events can be derived heuristically. Event patterns are implemented using event pattern languages (EPL) and event processing languages, see [3, p. 116-126]. In contrast to known event patterns, *unknown* event patterns can not be derived from heuristics based on an existing business process. They did not exist in the past or have not been recognized so far. An unknown pattern could be found with the help of event processing agents by analyzing the event cloud of an organisation and using specific algorithms to detect it, as described in chapter 2.

2. DETECTING UNKNOWN EVENT PATTERNS BY COMBINING DISCRIMINANT ANALYSIS AND NEURAL NETWORKS

The new fraud detection approach consists of a combination of discriminant analysis (see [4]) and neural networks (see [5]). This has the advantage, that every event represents one value as input for a neural network. The whole process is represented in fig. 1 and described below. The CEP engine creates event clusters on the base of known historical fraud events and no-fraud events. The total number of the clusters depends on how fine the event groups or clusters should be subdivided. The allocation of an event into a specific cluster depends on event attributes which are relevant for classifying an event as *fraud* or *no-fraud* event. By using the values of these relevant attributes for calculating the discriminant coefficient, the discriminant functions will be computed.

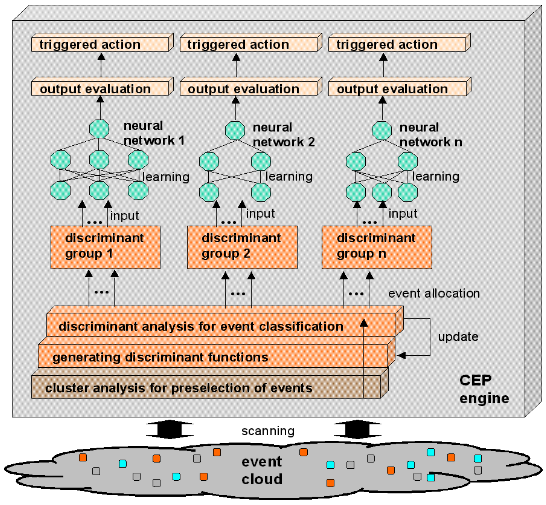

Figure 1: System architecture of combined discriminant analysis and neural network approach

The discriminant functions are used for allocating a new occurring event into a specific group of events. This is reached 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 base on the historic event clusters. This allocation process is defined exactly in [7]. The discriminant functions will be updated on the base of new discriminant group allocations after a defined time interval. So the discriminant functions keep dynamic for changing event occurrences and situations. At the beginning of the process, the global event cloud of an organization is scanned by a CEP engine. The events will be classified by inserting the relevant attributes in the discriminant functions and on the base of the results (discriminant value) they will be allocated into a specific discriminant group. On the one hand, an event can be allocated exactly to one specific discriminant group or on the other hand it can be a part of two or more discriminant groups. In that case, the discriminant value can be multiplied with a factor that represents the degree of membership to the discriminant group. This part of the process is described in [7]. For every discriminant group defined, a specific neural network is generated. The weights of the networks are determined by training them with discriminant values from known fraud and no-fraud event patterns of their specific discriminant group. So the discriminant values are used as input values for the neural networks. One discriminant value represents one event of a pattern that should be identified as fraud or no-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 find out whether the input events are a fraud combination or not. For known fraud combinations, the networks are trained with 1 as output value whereas known no-fraud combinations are trained with 0. In order to identify unknown combinations, a threshold is determined on the base of the training results e.g. 0.5. If the output value of an unknown input combination of events (respectively discriminant values) is greater than the threshold the system classifies it as fraud and reacts with a predefined action e.g. sending an alert to an operator. The values of a detected fraud pattern will be inserted in the training set which is used to train the network again, e.g. after the expiration of a predefined time interval just as one hour or one day. The 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]. The architecture described extends the work described in [7], because [7] defines the events inside a specific discriminant group as unknown pattern itself if the discriminant groups are defined exactly enough. In addition, the new approach uses neural networks for evaluating the occurring combinations of events allocated to a specific discriminant group of being an unknown fraud pattern or not.

3. EXPERIMENTAL RESULTS

The experimental environment is programmed in java by using Eclipse 3.2 as development tool. The java classes including the codes of the discriminant analysis algorithm and the neural network are embedded in StreamBase Studio [6] via a .jar file. This .jar file is connected with the java-operator component "FraudDetectionOperator. The event cloud is read into the java operator by EventInputAdpater and the results are written to a text file by using OutputFileAdapter. The parameters of the experiments are the following:

- Event cloud: 16 events (8 fraud and 8 no fraud events)
- Event-attributes: 2 fraud relevant attributes, 3 no-fraud relevant attributes
- Neural network initial weights :
  - -input node 1 to hidden node 1: -10.663
  - -input node 1 to hidden node 2: -5.280
  - -input node 2 to hidden node 1: -5.628
  - -input node 2 to hidden node 2: -2.589
  - -hidden node 1 to output node : -14.496
  - -hidden node 2 to output node : -6.416

After running the discriminant analysis as well as backpropagation learning with the event cloud and the test events, the algorithm delivers the following results:

- Disc. function: -0,0079 * x1 + 0,0101 * x2
- Critical discriminant value: 0.404
- Backpropagation loops: 20.000
- Learning factor: 0.9
- Neural network initial weights :
  - -input node 1 to hidden node 1: -34.025
  - -input node 1 to hidden node 2: 19.685
  - -input node 2 to hidden node 1: 18.563
  - -input node 2 to hidden node 2: -29.721
  - -hidden node 1 to output node : -19.913
  - -hidden node 2 to output node : -24.085

According to these experimental results, a fraud-dividing threshold of 0.4 can be determined for the created neural network. So if the activation value of the output node is greater than 0.4, the investigated known or unknown event pattern can be classified as fraud pattern. In this case, the application reacts with a predefined action e.g. sending an alert to the responsible operator. But this threshold of 0.4 can be adapted when the network has learned enough new patterns.

4. CONCLUSIONS AND FUTURE WORK

The work, presented in this paper, is still in progress. So by running these first experiments, the authors only want to show that the combination of discriminant analysis with neural networks is running successfully for a small set of events with two relevant attributes. In addition, the structure of the neural network only consists of two input nodes, two hidden nodes and one output node. Because of this simplified environment, the next steps are to extend the test and training data sets as well as the structure of the neural networks and the amount of historic events needed for creating the discriminant functions. The goal is to obtain research results about the new approach running in a more complex environment. In this context, it is also important to test the performance of the new approach in order to find out if it meets the requirements of real-time environments typical for CEP. A further step is to improve the experimental environment in such a way that it is able to simulate the structure of credit card transaction events and credit card frauds more exactly.

5. REFERENCES

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

[2] CEP Glossary. http://complexevents.com/?cat=15, downloaded 2006-12-06.

[3] Luckham, D. The power of events. Addison Wesley, San Francisco, New York, 2002.

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

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

[6] StreamBase Systems Inc. StreamBase Studio. http://www.streambase.com, downloaded 2007-10-31.

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