A Solution Architecture for Financial Institutions to Handle Illegal Activities: A Neural Networks Approach

Ashish Vikram,* Sivakumar Chennuru,* H. R. Rao,** Shambhu Upadhyaya+
*M&T Bank, **School of Management, SUNY Buffalo, +Dept. of CSE, SUNY Buffalo
avikram@ubbusiness.com, chennuru@ubbusiness.com, mgmtrao@buffalo.edu, shambhu@cse.buffalo.edu

Abstract

The banking and financial services industry today relies heavily on the use of networked computerized data systems to manage financial accounts and information on a real-time basis for millions of customers. This underlying technology is a source of a large quantity of information that can be used in the identification and prevention of financial fraud involving the illegal/unauthorized transfer of funds by entities external and internal to the victim financial institution. This paper develops a concept involving the use of neural networks to correlate information from a variety of technological and database sources to identify suspicious account activity.

1. Introduction

During the late 1980s and early 1990s, approximately 60% of the fraud reported by financial institutions to the FBI was related to bank insider abuse [1]. Though this form of fraud was subsequently overtaken by external fraud fuelled by technological advances and the availability of personal information through information networks, insider abuse “poses a serious threat to the integrity of the entire worldwide financial community” [2]. According to the FDIC, insider abuse (Fall 2001) accounts for all bank fraud and embezzlement cases investigated by the FBI [3].

1.1 Scenario

The banking and financial services industry today relies heavily on the use of networked computerized data systems to manage financial accounts and information on a real-time basis for millions of customers. Checking accounts, savings accounts, credit/loan accounts, trading accounts, business accounts etc. are all accessible and available digitally. Electronic banking allows customers to access their accounts and information over the World Wide Web, and some banks, financial institutions, and providers of financial services exist purely in an online form, without possessing any physical branches at all [Ex: www.x.com, www.netbank.com, www.compubank.com etc.].

Computerized systems allow access to accounts for not just customers, but also to customer service representatives, branch personnel, account managers, financial institution employees etc. Access to these accounts can occur through different interfaces depending on the method of access and depending on the person who is accessing the account. In physical terms, this access can take two forms as shown in Fig. 1.

The local machine interface is rarely used, while the network interface is the most heavily used. The local machine interface is easily secured through the use of controlled administrative rights as well as physically securing the location of the database server. It is the interaction over the network interface that has to be controlled and monitored. Most instances of unauthorized activity on an account will occur through the use of unauthorized application access, unauthorized transactions or unauthorized modification of data through administrative access over the network interface.

Because almost all of the activity that occurs for any given account over its lifetime will happen over the network interface, this interface represents a single point of monitoring for that activity. Currently, banks do not log all the activity that may occur at the network level or the level of each of the different access points because of the difficulty and cost of analyzing the data that is thus collected on a continuous basis. Limiting factors are the volume of data and the need for human intervention in identifying suspicious activity based on access/usage logs at the level of either the access points or the network interface. Instead they use an audit trail and log all accesses only after being notified of suspicious activity on an account, i.e., during a post-incident investigation. The use of a post-incident investigation mechanism also means that there is no real-time reaction to an unusual activity, with the damage control taking place after the fact.

0-7695-2056-1/04 $17.00 (C) 2004 IEEE

In this scenario, the availability of a tool that allows the identification of unusual account activity as it is occurring will be useful if it requires minimal human intervention, provides a facility for identification of unusual activity patterns (including a self-learning feature) in logged activity data and provides a risk assessment of activity on a continuous scale, making it easy for users to take action depending on the severity of the suspect activity.

2. Prior Research and Current Methods

Current methods for fraud detection and prevention focus on (a) access control (using basics such as identity verification prior to transactions) (b) the analysis of individual or classes/categories of accounts based on transaction history. Different organizations also use different methods for fraud detection and prevention. For example, one method is to detect fraud based on account activity thresholding [4] [7] in terms of both universal thresholds for all accounts and account specific thresholds. Risk scoring transaction activities on accounts and applying thresholds to them is another technique used for fraud detection [5]. Current methods also focus on fraud prevention at the point of contact/sale at the merchants' end, using methods [6] such as Card Verification Method, Payer Authentication, Lockout Mechanisms, Negative and Positive lists, Fraud Rules and Risk Scoring for credit card accounts. Analyzing individual transaction behavior for each account, building behavior profiles for each account [7] and then comparing current transactions with the profile is also a technique that is used in fraud detection.

Cahill, Lambert, et. al have suggested a signature based method to identify fraudulent transactions on telecom accounts [4]. Their method creates a probabilistic signature (essentially, profiling) separately for every single account based on past history and takes into account a number of variables that are considered important to identifying fraudulent transactions. Their approach permits pinpointing fraud in a single account (rather than aggregated data) and can also permit a near real-time detection of undesirable activity. However, the profiling process is very complex and the addition of any new variable into the system involves a major rework in the sense that the distribution of the new variable for each account has to be determined and also its interaction with the variables already present.

Verisign recently tested and implemented a Fraud Protection Service that combines geo-location data with data on timing of online credit card transactions as well as fraud filters, password management, allowed IPs, security audits, account monitoring and buyer authentication to detect and reduce fraud rates [8]. Verisign has previously used neural networks to detect fraud based on account activity and buying patterns [9]. Another company ClearCommerce also uses geo-location data to detect fraudulent transactions [10]. These examples indicate that geo-location information based on IP-addresses can be important to the process of detecting fraud.

Brockett et. al. [11] have applied Kohonen Networks and feature-map derived feed-forward neural networks in classifying Bodily Injury claims in terms of their suspicion level, and they also showed that the results using these neural-network based approaches were superior to using manual methods of fraud identification in Bodily Injury claims. Senator et. al. [12] have reviewed the architecture and function of FINCEN's AI based fraud detection system that uses a mix of machine AI and human ability to identify suspiciousness levels of activities in the records of large transactions that may be related to money laundering activities. Fawcett and Provost [13] have used a combination of rule-based methods and classifier neural networks as part of an adaptive system that detects cellular cloning fraud (superimposition of illegal or fraudulent activity on a legitimate account) based on a database of cellular call records. Their system is based on profiling and monitoring each customer's typical behavior, and analyzing massive amounts of cellular call data to determine general patterns of fraud.

Burge and Shawe-Taylor [14] use a recurrent neural technique and statistical behavior profiles with differential analysis covering both short and long term pasts to detect fraud in mobile phone usage. They apply unsupervised learning to eliminate the use of fraudulent examples required for training. They expect that their system will form part of a larger fraud detection suite that will include a rule-based fraud detection tool and a neural network fraud detection tool that uses supervised learning on a multi-layer perceptron.

These methods focus on applying various techniques to analyzing and preventing fraud on accounts from the point of view of external threats to an account. Given that insider abuse is a significant issue [3], our proposed solution takes a systemic view of the accounts in care of a financial institution and that can be applied to internal threats as well as external threats depending on the amount of information made available to it. Given that Internet/networking technology eliminates geographical and physical barriers in the commission of financial crimes, using information available with respect to the technological details of underlying transaction-facilitating technology used in the commission of crimes is important, and that is what our proposed solution includes.

3. Proposed Solution

The solution proposed here will focus on a financial account as the core around which the ideas are constructed. These ideas can be translated / adopted to other types of financial information/data that needs to be monitored for unusual activity.

This solution is based on the premise that there is a relationship between the predictor variables related to an account (transaction activity on an account, access activity on the account, etc.) and the predicted variable(s) (risk of fraud occurring on the account, degree of unusual activity on an account). Time is an important variable taken into account indirectly by using data accumulated over a period of time. The complexity imposed by the number of variables and the range of values they can take makes it very difficult and expensive, not to mention time-consuming, for an institution to develop rule-based (algorithmic) methods to model the relationship between the predictor variables and the predicted variables. This document proposes a speedier solution based on the ability of neural networks to quickly recognize underlying complex patterns in data and use them to assign risk levels to particular sets of transactions and activities observed in the environment of a financial institution.

3.1 Solution Architecture

The first step, in stopping or catching unauthorized activity, is knowing what information to analyze. To obtain a comprehensive picture, data available from different sources must be collected and logged before it can be analyzed effectively.

To this end, data to be logged would be of the following types: Account info; Account activity (including browsing, changes etc.); Originating source of change (customer, employee, sysadmin etc.); Network info for change originator (including routing info, IP addr. etc.); Interface information for change originator; Info about payees (geographical spread, risk level of payees, destination accounts-paypal, other accounts within same institution etc.) and info about beneficiaries of transactions.

The sources of the data to be logged may be disparate, such as the financial database that contains the account, the routers that direct the traffic to the financial database, the applications that are used by the employees within the institution to access the accounts, the applications/tools used by the system administrators to access the machines that act as containers for the accounts and the web servers that provide an interface into the account for the customers. These sources of log information can be in physically different locations. There is a need to recover this information and return it to a central location in real-time, if the solution is to be used in real-time, or at scheduled intervals if the solution has to be used on a scheduled basis to examine transactions that occurred since the last scheduled vetting.

The platforms used for these components of the financial institutions' architecture to provide services to their customers and employees may be widely different. For example, the web-server may be a Linux machine running Apache, the financial database may be DB2 running on an IBM mainframe while the log database may be an SQL server. The log information generated by each of the components interacting with an account may need to be compiled into a form suitable for the next link in the Comprehensive Account Activity Monitoring and Analysis Tool (CAAMAT).

A suggested method to work around the differences in sources and forms of log data would be the use of agents on each of the platforms.

3.1.1Role of the collecting agent

The collecting agent would collect log data and convert it to a form suitable for storage in the log database, preferably with minimum modifications. Once the data has been collected and converted, it would be entered into the log database using the underlying network. A different agent would be used for each different platform and/or source of data, but they would all talk to the same Central Activity Log Database (CALDB).

3.1.2 High Level Architecture

The High Level Architecture for the CAAMAT is diagrammed in Fig. 2.

The Central Activity Log DataBase (CALDB) holds all the information, past and current related to the different categories of activity noted earlier.

The Data Interface provides a means of accessing the data within the CALDB. It may perform data conversion functions if necessary. Principally, it is a means of accessing the database over the underlying network, allowing the CALDB and the Activity Analyzer to be located separately.

The User Interface is the front-end of the solution. It provides access to the results produced by the activity analyzer, allows training of the solution, modifications and selection of the training data sets, customization of the various parameters used by the solution and maintenance and administrative functions.

The Response System Interface provides a means of linking the CAAMAT to the Management Information Systems of the institution using the solution. It may provide a link to the real-time information that can be produced by the solution through the use of 'alerts'. These can be linked to say the data center that monitors the health of the institution continuously. It can also be programmed to notify the right personnel through various communication media such as email and automated pager messages as part of risk / problem management procedures in place.

The core of the solution architecture is the Activity Analyzer. This component of the CAAMAT is itself a complex tool based on neural networks applied to the data in the activity log database. The activity analyzer has the internal structure diagrammed in Fig. 3.

3.2 Activity Analyzer

The Activity Analyzer is composed of the following parts:

1. Account Classification Engine (ACE)

This engine classifies each account into different grades dynamically based on the following formula:

Account Grade = Account rating x Activity level

The account rating is a combination of factors such as initial risk, income level of account holder etc. The purpose of this rating is to classify the account into a category that will define the broad range of values that may be expected to be seen in the variable parameters of this account.

The activity level is based on historical data over the past few time slots (periods), typically a couple of weeks or whatever time span is appropriate for this particular institution. Note that the same time span must be used for all accounts. It is assigned dynamically by the Account Classification Engine and includes current activity level information.

The purpose of the ACE is to categorize the accounts dynamically, mapping data into a fixed type of category so that the neural networks in the Processing Engine (PE) that will be monitoring a particular category see data that they has seen before—covering all eventualities and not requiring the neural networks to extrapolate results outside their range of experience. It also partially serves the purpose of offsetting changes in the operating environment of the institution using the solution such as changes in the activity level of accounts related to say, changes in the economy.

There is a separate processing engine for each account grade, with its internal networks trained on datasets that are peculiar to that account grade.

2. Holding Databases (real-time data, learning/re-training data)
   The holding databases are of two types:
   a. The *Real-time Database* that holds current log information to be processed by each PE for real-time analysis. This need not hold historical data prior to the time slot under consideration and
   b. The *Learning/Re-training Database* that holds log data approved by the user for training the neural networks in the processing engine. Typically, this database will hold data from many different time slots. The user selects the data that is clearly known to be related to fraudulent and clean accounts.

Here it is useful to note that the proposed solution has to be combined with operational procedures to retrain the PEs whenever the environment changes significantly, or on a periodic basis in a time frame suitable to the operating conditions of the institution using this solution.

3. Processing Engine (PE)

The PE is the module that performs the actual functions of monitoring the logged data and analyzing it for anomalies. All data in a set corresponding to an appropriate time slot as chosen by the user is assigned a risk grade ranging between limits set by the user.

There may be several Processing Engines in a given solution implementation. The number of PEs depends on the number of account grades that are set up by the user of the solution, with a separate PE being assigned to each account grade. The account grades are assigned to particular accounts by the Account Classification Engine (ACE) according to parameters that can be set by the user.

The PE has several sub-components that are described below.

   a. Scaling Function Module (SFM)

   The function of this module is to scale the data appropriately for use by the neural networks within the PE. This scaling is required because neural networks effectively accept numeric input that is meaningless out of its context and the input range has to be limited for effective performance. The SFM translates or maps the logged information into a numeric format that can be fed to the inputs of the neural networks. For example, IP addresses must be translated to an appropriate scale of numbers, account parameters such as gender of the owner will have to be mapped to a numeric representation that can be understood by the neural networks.

b. *Account / Payee / Source Activity Monitors*

These, along with the Unusual Activity Monitor, are the key to the entire proposed solution architecture. Together, they form the core of the processing engine that identifies and recognizes the relationship between the predictor variables that are logged by the system and the predicted variables that form the output of the system.

The Account, Payee, and Source activity monitors are unsupervised Kohonen neural networks that have a Self-Organizing Feature Map. Unlike supervised neural networks that are trained through data sets that contain input variables as well as associated outputs, these unsupervised networks use data sets that contain only inputs and are used for exploratory data analysis. The inputs to each type of these activity monitors are different.

The Account activity monitor receives inputs related to the activity performed on the accounts by all entities accessing the accounts. The Payee activity monitor receives inputs related to the nature and type of the beneficiaries of the transactions occurring on the accounts. The Source activity monitor receives inputs related to the nature, location and mode of access of the source of the activities on the account.

These neural networks are trained on data sets approved by the user and classify the inputs into categories for each of the activity monitor types. Because there may be insufficient data sets available to model accurately fraudulent or suspicious transactions, the problem is turned on its head and these neural networks are trained on “good” data sets, i.e., they are taught to recognize patterns related to known good transactions— a significant majority of transactions are such and can thus be conveniently used to train the networks.

These neural networks reduce the n-dimensional data inputs (the number of variables is large) to a 2-dimensional output that produces numeric patterns related to complex patterns observed within the logged data. These neural nets are trained directly on the data from the learning/re-training database. Periodically, the neural nets are re-trained to reflect changes in the operating environment of the institution.

The output of these trained Kohonen networks is the input to the next stage, along with raw data from the SFM that was not used by the activity monitors in this stage. The output is used to train the UAM stage that follows, while the output of the trained Kohonen nets based on current or real-time data is fed to the trained UAM in the operational phase.

c. *Unusual Activity Monitor (UAM)*

The UAM is a Multi-Layer Perceptron type of Neural Network that is trained to minimize the error in recognizing a “good” transaction or data set. This type of network consists of neural units that each perform a biased weighted sum of their inputs and pass this activation level through a transfer function to produce their output. The units are arranged in a layered feed-forward topology. Training data sets are used to adjust the weights and thresholds of the units to minimize the error or the difference obtained by comparing the actual output with the desired output. Since the outputs have to be specified along with the inputs, these types of neural nets need supervision during the training phase. The type of training algorithm that can be used is Back-Propagation.

The inputs to the UAM are the patterns generated by the Account/Payee/Source activity monitors as well as data from the SFM that was not fed into the A/P/S activity monitors. Again, the UAM is trained to recognize “good” data sets and assigns a score (or classification) based on the matching of the input patterns with the function modeled by the UAM. Its output is fed to the post processing stage that converts information into a form suitable for use in decision support systems.

An alternative network type for the UAM stage would be the Radial Basis

Function (RBF) Network. Depending on the sophistication of the ACE, using an RBF network will result in better or poorer performance. RBFs are easier to design and train, but they have difficulty in extrapolating results beyond the boundaries of their training sets.

d. Post Processing Stage

This stage is required to convert the output of the UAM, which assigns a score based on how good a pattern matches the modeled function, into a score that reflects the degree of risk or suspicion associated with a particular data set or transaction. Because the neural networks were trained to recognize "good" data sets, this stage essentially performs an inversion of the assigned score to reflect the degree of unusual activity.

The self-organizing Kohonen networks are used to identify underlying patterns in the input data and also serve to reduce the dimensionality of the problem that has a large number of inputs or variables. The proof-of-concept for this application of Kohonen Networks is included in the Appendix. The second stage network (UAM) then matches their output to recognize good or bad transactions and data sets.

Choice of the size of the neural networks and the number of layers in them has to be made carefully depending on the expected complexity of the data sets. This will be similar for most financial institutions, differing based on the diversity of their customer base. Inevitable, some degree of trial and error will be involved during the first set up.

4. An Example of the use of CAAMAT

Consider the following scenario: A consumer, based in New York City, has a checking account with a bank. This checking account is associated with an Internet Banking Account that allows the user to transfer funds between his checking account and savings account with another bank. The Bank's back-office technology operations are in Buffalo, NY, so the database holding the account is in Buffalo, NY. This user regularly transfers funds between these accounts using Internet banking. Under normal conditions, the flow of data related to this account would look like this:

1. A transfer occurs on this account, from this account to another account with another bank or to another account with the same bank.
2. Associated with this particular transfer are the IP addresses of the user, and the transaction is performed through the web-banking server of the host bank (say Bank #1).
3. A sysadmin within Bank#1 regularly scans accounts for audit purposes. So the logs reflect this access of the accounts by the sysadmin.
4. Associated with the sysadmin's access are the IP addresses of the sysadmin's machines.

A neural network trained on these "good" transactions would now recognize the pattern representing the account transfers with the general range of the user's Internet IP addresses, and just auditing accesses with the sysadmin's range of IP addresses. If the geographical information was made available to the neural network, it associates the user's normal geographical region with account transfers and the Bank's internal IP addresses as well as the Buffalo NY geographical location with audit accesses on the account.

Though the neural network may not retain the exact details of the transactions above, it does identify the general pattern that valid user account transfers occur through the web-banking interface from a general geographical location outside the bank and outside the firewall protecting the bank's systems, and the pattern that audit accesses (which do not result in money transfers) occur from within the bank and within the bank's firewall.

Now, if the sysadmin "goes bad", having access to the user's accounts, he/she one day attempts to transfer money to his/her own account in another bank. This is attempted from within the bank. The trained neural network that is monitoring the transaction in real-time realizes that this is an unusual pattern, because the transaction flow is occurring through the web-server, but the input to the web-server is from within the Bank itself. The neural network then flags this unusual activity and triggers an alarm or logs it into a list of "items to audit" by the next available customer representative, depending on the account grade assigned to the account by the ACE (Account Classification Engine).

Again, if the sysadmin just notes down the username and password for the account and attempts to do the transfer from outside of work hours, the geographical location of the sysadmin flags the account transfer as unusual. If the user makes transfers only during the day (official account) and the sysadmin attempts the transfer during the evening, this would set off the red flags too.

A third possibility of catching the transaction's "unusual" nature is if the transfer is occurring to a destination that is different from the user's normal set of account destinations (the sysadmin's destination account bank is different from the user's destination bank accounts).

The use of CAAMAT has potential for detecting fraud occurring outside the bank as well. For example, if the user's credit card account is used to pay Internet based companies regularly, an association exists between the destination accounts (which are corporate accounts), and the user's credit card account. If a hacker obtains the credit card account and attempts to transfer funds to his/her own account (presumably a private account), the neural network monitoring the transaction would flag this as an unusual activity.

In these types of scenarios, a bank would under normal circumstances find it difficult and expensive to monitor thousands of accounts for such unusual activity on a 24/7 basis, which is why post-incident measures are used to track down fraudulent transactions. A bank would come to know of the fraudulent transaction only through notification by the user who is missing funds. If the fraudulent transactions go unnoticed by the user for a long period of time, then the customer loses out. However, with the use of CAAMAT, it would be possible to actually monitor all the accounts for unusual activity 24/7, minimizing the cost to the bank and its customers in terms of fraudulent transactions. The success of CAAMAT depends on its ability to form associations between bits of information available from many different sources (the more the information available, the easier it is for CAAMAT to identify unusual activity)—normally, performing this correlation manually would require an enormous amount of time and resources, especially for thousands of accounts. CAAMAT brings to this scenario the speed of electronic automation & continuous 24/7 operations, and the intelligence or human brain-like ability to recognize associations between diverse pieces of information- all made possible through the use of artificial neural networks (that mimic the neural networks inside the human brain) combined with the data gathering and processing capability of networked digital computers.

5. Conclusion

The use of a layered approach- using rule-based classification, unsupervised self-organizing neural networks as well as supervised neural networks trained to assess the degree of risk associated with a particular (set of) transaction(s)- for monitoring and analyzing account transactions at financial institutions taking into consideration information available from disparate sources within the financial system should help alleviate the difficulty and cost of continuous real-time monitoring of account activity. Ultimately, the goal is to catch unusual activity, and by association, fraudulent activity that exacts a high toll from financial institutions in an era of computerized financial systems that are exposed to undesirable elements both within and outside the institution.

Scalability, performance and reliability are important characteristics that determine the usability of a system in the real world. Our paper does not examine these aspects of CAAMAT and this has been left as a topic for future study. It would be useful to note, however, that individual components of the system such as the use of neural networks have been successfully applied to fraud detection previously and in current systems. In our approach, the scope and number of variables being examined simultaneously has been increased. Qualitatively speaking, increasing the complexity of the problem by increasing the number of variables being interpreted can be tackled by using more sophisticated neural networks with a greater number of neurons and / or layers at the expense of greater computational power. Because the neural networks can be replicated without affecting the accuracy and because each network only needs to look at one transaction at a time in the detection phase, clustering methods (computing clusters running several instances of the neural networks) can be used to handle scale.

Additional research would be also required to create methods to convert different data sources related to an account into formats suitable for use by the neural networks. Some of these tasks would be relatively easy, such as converting source IP addresses related to a transaction into a format readable by neural networks, which could be accomplished by feeding network traffic trace files or packets related to an account/transaction to a filter that would extract the needed information and store it in a database. More difficult is the task of associating a particular trace/network packet with a particular account or transaction. Similar issues arise in associating and feeding data from administrative server logs related to a particular account and transaction to the neural networks.

A (methodological, rather than technical) limitation of our approach is that it looks at each transaction in context of its variables, but not within context of sequences of transactions over time other than what is implied in the prior training of the network itself. This feature of being able to look at sequences of transactions over time is a desirable feature that can improve the accuracy of fraud detection. The drawback of such a feature would be the need to maintain a history of transactions for each account in memory, either in the form of specific neural maps for each account or the entire sequence of transactions for every account being monitored. This would lead to increased memory and computational requirements, and would possibly have to be done offline rather than within a short period of time of a transaction occurring. Our current approach does utilize sequential information related to a particular account, but in the training phase for the detector neural nets, hence not explicitly for each individual account, only for classes of accounts.

Appendix:

The proof of concept was implemented using a Kohonen network [15] to recognize the variation in patterns as the nature of transactions change. The network was fed with the transactions representing good transactions and the patterns were recorded for the good transactions. Each set of transaction consisted of a signature representing the source IP of the request, the destination IP of the request, the account number, the amount of transfer requested, the time of transaction, the time interval between the current and the last transaction. The patterns that were tested included the variation in amounts of transaction other parameters being same (case I), the variation in time interval (dormant accounts being activated with large amount withdrawals) (case II), the variation in the source and destination machines for the same account (case III) and account which exhibited random behavior over time (case IV). 8400 transactions representing these behaviors were fed into the network to recognize the patterns. The training for these transactions was completed in about 17 seconds. The patterns that were generated for these transactions were recorded. The trained network was tested to identify the patterns representing normal and abnormal transactions. These patterns were tested against a good transaction and a bad transaction representing each case. These represent the transactions that will be coming through in the real time system. The trained networks were compared to see if there are any differences in the target output map between the good transactions and the bad transactions. The output maps that are obtained can be compared by using the squares of the differences in the activation levels of each unit in the output map. The larger the value of the squared weight differences, the greater the difference in the transactions and vice versa.

The outputs maps that were obtained for Case I mentioned above are given in Fig. 4.

Similar output maps were obtained for the remaining cases showing clear distinction in the activated regions for the changes in the nature of transactions. However, the amount of change differences being shown for the transactions varied in each of the cases with the maximum difference being shown for variation in multiple parameters. For the case of the random behavior of the account, no single discernible pattern was observed and the output map varied for each transaction. For the transactions of this type, the output measures can be based on an aggregate output map values obtained over the multiple transactions.

The Proof-of-Concept will be extended to cover more possible patterns of transactions and the discovery of fraudulent transactions from the output of the kohnen network will also be automated in the future.

References

1. http://www.fbi.gov/hq/cid/fc/fifu/about/about_fifu.htm , May 14, 2003
2. Kaiser, J., Internal Threats to Financial Institutions (Insider Abuse), http://www.bankersonline.com/articles/sfpv02n08/sfpv02n08a9.html , May 14, 2003 Originally appeared in Bank Security & Fraud Prevention, Vol. 2, No. 8, 8/95
3. http://www.osbckansas.org/DOB/DOBQuarterlyInterest/Fall01/Insider.html May 18, 2003
4. M.H. Cahill, D. Lambert, J.C. Pinheiro,D.X. Sun, Detecting fraud in the real world, In: Handbook of Massive Datasets. Kluewer. 2000, http://cm.bell-labs.com/cm/ms/departments/sia/doc/hmds.pdf
5. Banks streamline costs with automated customer messaging, http://www.csc.com/features/2002/55.shtml May 21, 2003.
6. Fraud Prevention Guide, ClearCommerce White Paper, http://www.techieindex.net/whitepapers/pdf/ClearCommerce_Fraud_Prevention_White_Paper.pdf
7. SearchSpace and IBM help banks combat fraud and money laundering,

http://www.searchspace.com/library/searchspace_ibm_totalstorageproven_flyer_us.pdf, May 21, 2003

8. Fraud Protection Services, Verisign, Sept. 28, 2003 http://www.verisign.com/products/payflow/fraud/protection/faq.html#1a http://www.verisign.com/products/payflow/fraud/protection/accountmonitoring.html http://www.verisign.com/resources/datasheet/fraudprotection/CyberDefenseSys_Sheet.pdf
9. Verisign introduces new credit card fraud screening services for online merchants, Aug 2, 2000 http://www.verisign.com/corporate/news/2000/ehnc.html
10. ClearCommerce announces GeoLocator, Oct. 23, 2001 http://www.clearcommerce.com/press/GeoLocator_Release_10-23.html
11. P.L. Brockett, X. Xia, R.A. Derrig, Using Kohonen's Self-Organizing Feature Map to Uncover Automobile Bodily Injury Claims Fraud, The Journal of Risk and Insurance, 1998, Vol 65, No. 2, 245-274
12. The Financial Crimes Enforcement Network AI System (FAIS) – Identifying Potential Money Laundering from Reports of Large Cash Transactions, Senator et. al., AI Magazine,16(4): 21-39, 1996.
13. T. Fawcett, F. Provost, Adaptive Fraud Detection, Journal of Data Mining and Knowledge Discovery, v.1.n.3, 1(3): 291-316, 1997
14. P. Burge, J. Shaw-Taylor, An unsupervised neural network approach to profiling the behavior of mobile phone users for use in fraud detection, Journal of Parallel and Distributed Computing, Volume 61, Issue 7, 915-925, 2001
15. LENS, Neural Network Simulator, http://tedlab.mit.edu/~dr/Lens/

Figures

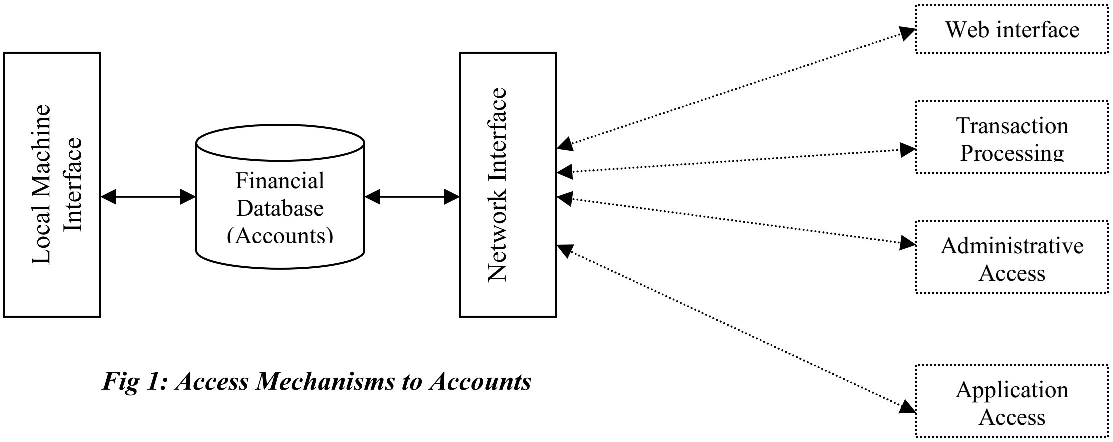


Fig 1: Access Mechanisms to Accounts

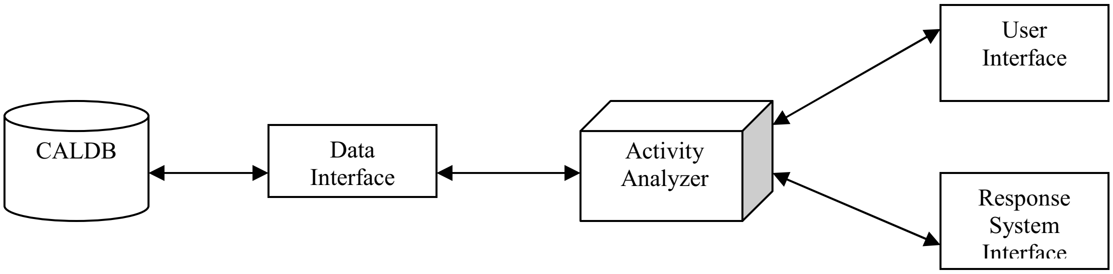


Fig 2: Comprehensive Activity Monitoring & Analysis Tool (CAAMAT)

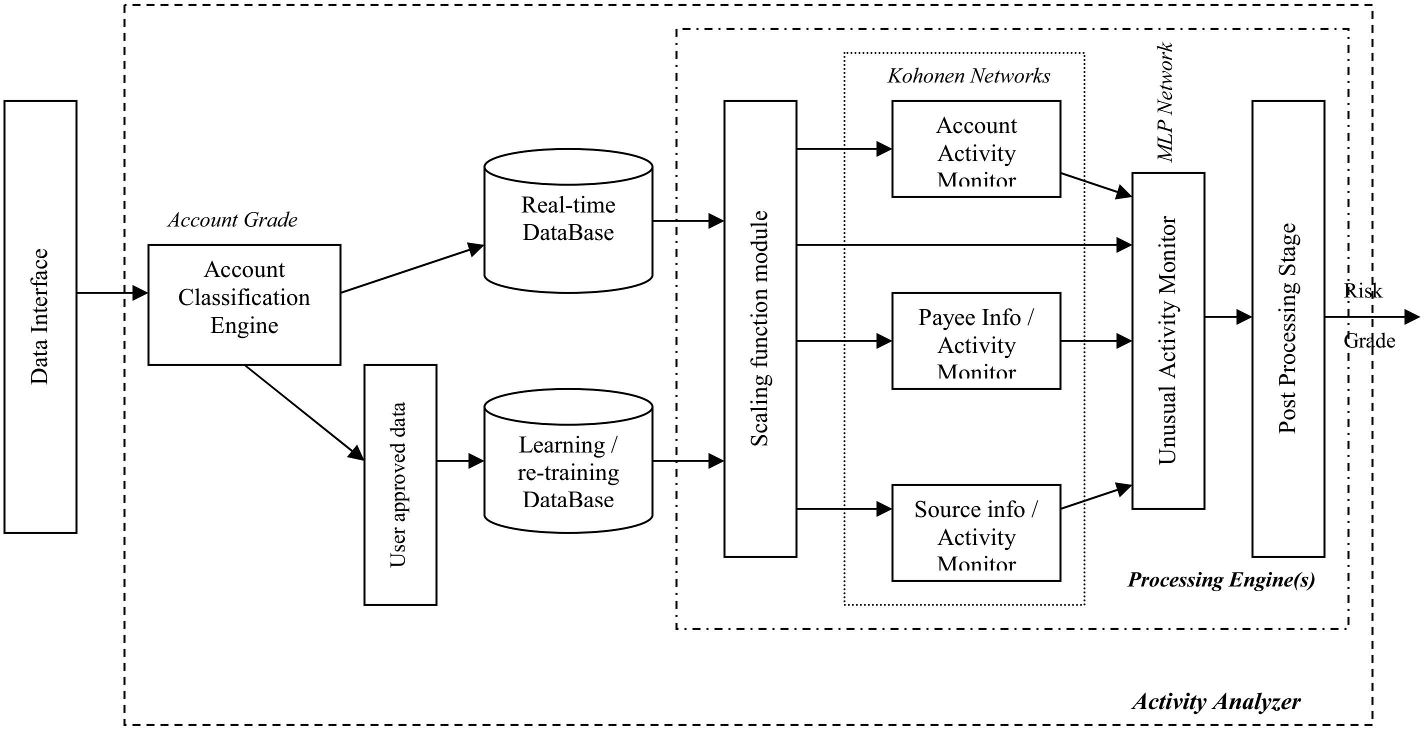


Fig. 3: Activity Analyzer: Internal Architecture and Components

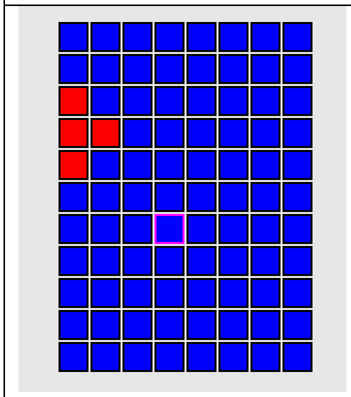
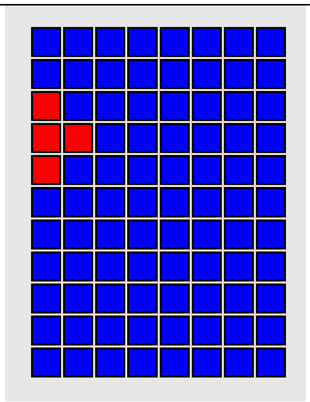
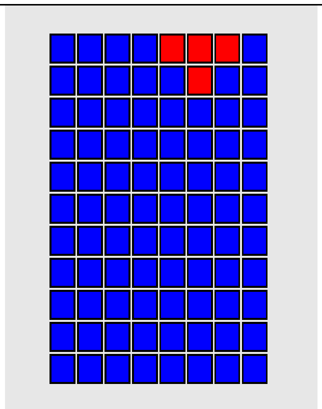
Training Output Map	Testing Output Map for similar transaction	Testing output map for abnormal transaction
		

Fig. 4: Kohonen Output Maps for Fraudulent Case I